

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:
Ingenieros de Sistemas

TEMA:
DISEÑO E IMPLEMENTACIÓN DE CONTROL DOMÓTICO (CON SISTEMAS
EMBEBIDOS) PARA CONECTARSE CON APLICACIONES ADAPTIVAS
BASADOS EN IOT

AUTORES:
JUAN DIEGO FONSECA YUPA
DANIEL ANDRETTY SORIA BADILLO

TUTOR:
MANUEL RAFAEL JAYA DUCHE

Quito, agosto del 2020

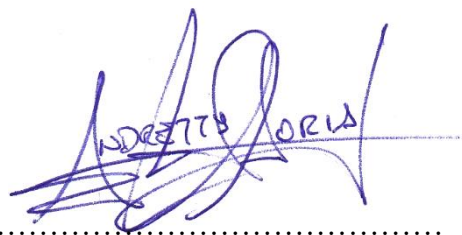
CESIÓN DE DERECHOS DE AUTOR

Nosotros: Juan Diego Fonseca Yupa, con documento de identificación N° 1715729305 y Daniel Andretty Soria Badillo, con documento de identificación N° 1722722392, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “DISEÑO E IMPLEMENTACIÓN DE CONTROL DOMÓTICO (CON SISTEMAS EMBEBIDOS) PARA CONECTARSE CON APLICACIONES ADAPTIVAS BASADOS EN IOT”, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en digital a la Biblioteca de la Universidad Politécnica Salesiana.



JUAN DIEGO
FONSECA YUPA
CI: 1715729305



DANIEL ANDRETTY
SORIA BADILLO
CI: 1722722392

Quito, agosto de 2020

DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “DISEÑO E IMPLEMENTACIÓN DE CONTROL DOMÓTICO (CON SISTEMAS EMBEBIDOS) PARA CONECTARSE CON APLICACIONES ADAPTIVAS BASADOS EN IOT”, realizado por Juan Diego Fonseca Yupa y Daniel Andretty Soria Badillo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto del 2020



MANUEL RAFAEL JAYA DUCHE

CI: 1710631035

DEDICATORIA

Al culminar una de las etapas más importantes de mi vida, vienen muchos recuerdos a mi mente, donde aparecen personas que me acompañaron durante todo este grandioso camino. Por ese motivo quiero dedicar este documento en primer lugar a Dios, por permitirme obtener experiencias inolvidables durante mi período universitario, por acompañarme en todos los momentos más difíciles de mi vida para no desfallecer, al igual que en esos momentos de felicidad que he tenido.

A mi mamita Julie, quiero dedicarle, porque día a día sus consejos han sido indispensables, porque siempre ha sido mi inspiración para seguir adelante, porque es la persona que mira por mi bienestar antes que el suyo, porque siempre me brinda su sabiduría y amor, porque mi vida y mis logros son gracias a usted, gracias mamita amada por regalarme la mejor herencia que un hijo puede tener.

A mi papito Guido, porque supo enseñarme desde muy pequeño como enfrentar cada obstáculo que me permitió crecer, porque siempre me apoya así no conozca del tema motivándome a investigar más. También porque sin usted este proyecto no sería posible, fue irremplazable día y noche sacrificando todo junto a mí. Esto es para usted papito, lo amo con toda mi alma, mi superhéroe.

A mi querido hermano Cris, si bien es cierto tienes un camino académico muy diferente al mío, pero siempre estuviste apoyándome y acompañándome cuando necesité de ti, demostrándome que no hay amor más grande que el de la familia, además, recuerda que nacimos para aprender y ser mejores día a día.

Daniel Andretty

DEDICATORIA

En primer lugar, dedico este proyecto técnico a Dios por haber guiado cada uno de mis pasos para llegar hasta donde estoy, por permitirme disfrutar de mi familia, adquirir conocimientos y experiencias en mi vida universitaria, cuidando de mí y de toda mi familia.

A mi madre Marlene, pues ella es el principal cimiento en lo que es la construcción de mi vida profesional, por brindarme su confianza y jamás dejar de creer en mi capacidad, para forjar y dejar un legado para las futuras generaciones.

A mi padre Washington, mi tía Irene, mi tío Mauricio y demás tíos que siempre estuvieron y están allí para darme una mano cuando más lo necesito.

A mis primos Cristian, Fernando, Santiago y Lida, que con sus ejemplos pude llegar a obtener mi título profesional, ellos con sus aciertos y errores me enseñaron como forjar un camino y tener la fortaleza para llegar a cumplir la meta.

A mis pequeñas Luciana, Fiorella, Kaytlin, Cristina y mi pequeño Axel, que con sus ocurrencias dan sentido a mi diario vivir, por ellos mejoro día a día, para que tengan un mejor porvenir y otro ejemplo a seguir.

A mis amigos Daniel, Santiago, Andrés, Iván, Geovanny y Juan que supieron brindarme su apoyo cuando estaba a punto de darme por vencido y que con sus consejos ayudaron a levantarme y seguir en pie de lucha para conseguir este objetivo.

Juan Diego

AGRADECIMIENTOS

Agradecemos a la Universidad Politécnica Salesiana que nos ha formado académicamente y nos brindó ese extra referente a los valores y principios que todo ser humano debe tener; a nuestro tutor de titulación el Ingeniero Rafael Jaya por habernos guiado pacientemente durante cada paso de este proceso.

A usted Inge Julio, porque supo aconsejarnos cuando las opciones investigadas entregaban resultados no tan buenos.

A Verito, por ser una persona incondicional que nos orientó con su conocimiento para motivarnos a culminar exitosamente este proyecto técnico.

Finalmente, a nuestros amigos, familiares y todos aquellos que de alguna manera aportaron con un granito de arena para alcanzar este objetivo de convertirnos en ingenieros de sistemas que tanto anhelamos.

Daniel Andretty Soria Badillo

Juan Diego Fonseca Yupa

Índice

Introducción.....	1
Problema	1
Justificación	2
Objetivos	3
Marco Metodológico	3
1. Marco Teórico	5
1.1. Internet de las Cosas (IoT)	5
1.2. Sistemas embebidos.....	5
1.3. Domótica	6
1.3.1. Viviendas existentes.	6
1.3.2. Preservación de la naturaleza y ahorro energético.	6
1.3.3. Desarrollo de comunicaciones.....	6
1.4. Arduino	7
1.5. Hardware	7
1.5.1. Tarjeta de desarrollo ESP-8266.....	7
1.5.2. Tarjeta de desarrollo ESP-32 DevKit3.	7
1.5.3. Cámara de video ESP32 CAM.	8
1.6. Módulos	8
1.6.1. Sensores.....	8
1.6.1.1. Red de sensores (WSN).	8
1.6.1.2. Sensor detector de movimiento pasivo infrarrojo HC-SR501 (PIR).	8
1.6.1.3. Sensor de temperatura/humedad DHT11.	9
1.6.1.4. Sensor de gas y humo MQ6.	9
1.6.1.5. LDR.....	10
1.6.2. Actuadores.....	10
1.6.2.1. Diodo emisor de luz (LED).....	10
1.6.2.2. Motor modelo 28BYJ-48 paso a paso.....	10
1.6.2.3. Módulo integrado para motor ULN2003.	11
1.7. Software.....	11
1.7.1. Arduino IDE.	11
1.7.2. PHP.....	11
1.7.3. Message queue telemetry transport (MQTT).	11

1.7.4. MQTT – SN.....	12
2. Análisis Hardware - Software	13
2.1. Análisis de Hardware.....	13
2.1.1. Representación gráfica del prototipo experimental.	13
2.1.2. Análisis tarjeta de desarrollo ESP-8266.	14
2.1.2.1. ESP-01.	14
2.1.2.2. Wemos mini D1.	15
2.1.2.3. NodeMCU v1.0 / V3.	16
2.1.2.4. Análisis de priorización para placas ESP-8266.	16
2.1.3. Análisis tarjeta de desarrollo ESP-32.	21
2.1.3.1. ESP-WROVER-KIT-VB.	21
2.1.3.2. ESP32-LyraT.	21
2.1.3.3. ESP32-DevKitC.	22
2.1.3.4. Análisis de priorización para placas ESP-32.	23
2.1.4. Análisis sensor de presencia.	25
2.1.4.1. Sensor PIR.....	25
2.1.4.2. Sensor de movimiento por Radar HB-100.	26
2.1.4.3. Sensor ultrasónico HC-SR04 P.	26
2.1.4.4. Análisis de priorización para sensor de presencia.	27
2.1.5. Análisis sensor de temperatura/humedad.	29
2.1.5.1. Sensor DHT11.....	29
2.1.5.2. Sensor DHT22	29
2.1.5.3. Análisis de priorización para sensor de temperatura/humedad.....	30
2.1.6. Análisis sensor de gas/humo.	32
2.1.6.1. Sensor MQ-2.....	32
2.1.6.2. Sensor MQ-5.....	33
2.1.6.3. Sensor MQ-6.....	33
2.1.6.4. Análisis de priorización para sensor de gas/humo.	34
2.1.7. Análisis motor para movimiento de puerta y persiana.	36
2.1.7.1. Motor modelo 28BYJ-48 paso a paso.....	36
2.1.7.2. Motor DC básico 5 voltios.	37
2.1.7.3. Servomotor SG-90.	37
2.1.7.4. Análisis de priorización para motor de puerta y persiana.	38
2.1.8. Análisis cámara de video.....	40
2.1.8.1. ESP32-CAM.	40
2.1.8.2. ESP-EYE.....	41
2.1.8.3. Análisis de priorización para cámara de video.	41
2.2. Análisis de Software	44
2.2.1. Análisis software para programación de código.	44
2.2.1.1. Arduino IDE.....	44
2.2.1.2. Arduino Create.....	44
2.2.1.3. ESPlorer IDE.....	44
2.2.1.4. Análisis de priorización para software utilizado en desarrollo de código.	45

3.	Diseño y Desarrollo	48
3.1.	Configuración General para Programación de Arduino.....	48
3.1.1.	Librerías para programación del código.....	51
3.2.	Diagramas para Módulos dentro del Hogar	52
3.2.1.	Dormitorios.....	52
3.2.1.1.	Código Dormitorios.	53
3.2.2.	Comedor - cocina.	55
3.2.2.1.	Código comedor - cocina.	55
3.2.3.	Sala - pasillo.	56
3.2.3.1.	Código sala - pasillo.....	56
3.2.4.	Pasillo con cámara ESP32-CAM.....	57
3.2.4.1.	Código pasillo con ESP32-CAM.	57
3.2.5.	Garaje - baño.	59
3.2.5.1.	Código garaje - baño.	59
3.3.	Diagrama Lógico y Físico de Bloques de Hardware	60
4.	Periodo de Pruebas.....	63
4.1.	Pruebas de Operatividad	63
4.1.1.	Funcionamiento de elementos.	63
4.1.2.	Placa ESP y MySQL, php, get/post, Arduino IDE.....	64
4.1.3.	Placa ESP y WebSockets, php, Arduino IDE.....	66
4.1.4.	Placa ESP y MySQL, Arduino IDE.	68
4.1.5.	Placa ESP y MQTT-SN, Arduino IDE.....	69
4.2.	Cuadros Comparativos de Resultados	72
4.3.	Confiabilidad de Datos Enviados y Recibidos con Protocolo MQTT-SN	75
4.4.	Optimización de Dispositivos	77
4.5.	Análisis de Consumo Energético para Focos del Hogar	78
4.6.	Prototipo Experimental	85
4.7.	Vinculación de Hardware con Software.....	88
4.7.1.	Encendido de focos (actuador) entre hardware y software.	88
4.7.2.	Apertura y cierre de persiana (actuador) entre hardware y software.....	89
4.7.3.	Streaming y captura de imagen con placa ESP32-CAM desde el hardware.....	91
4.7.4.	Alerta por detección de movimiento (sensor) entre hardware y software.....	93
4.7.5.	Notificación de temperatura (sensor) entre hardware y software.....	94
4.7.6.	Alerta por detección de gas (sensor) entre hardware y software.....	96
	Conclusiones	97

Recomendaciones	99
Lista de Referencias	100

Índice de Tablas

Tabla 1. Características para opciones de placa ESP-8266.....	17
Tabla 2. Ponderación de criterios	17
Tabla 3. Criterios y Opciones.....	17
Tabla 4. Criterios de Priorización	18
Tabla 5. Matriz Precio	18
Tabla 6. Matriz Capacidad de Procesamiento	19
Tabla 7. Matriz Nivel de Respuesta	19
Tabla 8. Matriz Disponibilidad	20
Tabla 9. Matriz Criterios vs Opciones	20
Tabla 10. Características para opciones de placa ESP-32.....	23
Tabla 11. Criterios y Opciones.....	24
Tabla 12. Criterios de Priorización	24
Tabla 13. Matriz Precio	24
Tabla 14. Matriz Capacidad de Procesamiento	24
Tabla 15. Matriz Nivel de Respuesta	25
Tabla 16. Matriz Disponibilidad	25
Tabla 17. Matriz Criterios vs Opciones	25
Tabla 18. Características para opciones de sensor de presencia	27
Tabla 19. Criterios y Opciones.....	27
Tabla 20. Criterios de Priorización	28
Tabla 21. Matriz Tiempo de Detección.....	28
Tabla 22. Matriz Rango de Detección.....	28
Tabla 23. Matriz Precio	28
Tabla 24. Matriz Tiempo de Inicialización	28
Tabla 25. Matriz Criterios vs Opciones	29
Tabla 26. Características para opciones de sensor de temperatura/humedad	30
Tabla 27. Criterios y Opciones.....	31
Tabla 28. Criterios de Priorización	31
Tabla 29. Matriz Precio	31
Tabla 30. Matriz Rango de Detección.....	31
Tabla 31. Matriz Precisión de Medición	32
Tabla 32. Matriz Tiempo de Muestreo.....	32
Tabla 33. Matriz Criterios vs Opciones	32
Tabla 34. Características para opciones de sensor de gas/humo.....	34
Tabla 35. Criterios y Opciones.....	34
Tabla 36. Criterios de Priorización	35
Tabla 37. Matriz Precio	35
Tabla 38. Matriz Rango de Detección.....	35
Tabla 39. Matriz Gas Característico.....	35
Tabla 40. Matriz Tiempo de Precalentamiento	36
Tabla 41. Matriz Criterios vs Opciones	36
Tabla 42. Características para opciones de motor de puerta y persiana.....	38
Tabla 43. Criterios y Opciones.....	38
Tabla 44. Criterios de Priorización	39
Tabla 45. Matriz Precio	39

Tabla 46. Matriz Rotación.....	39
Tabla 47. Matriz Alimentación	39
Tabla 48. Matriz Precisión & Control de Movimiento	40
Tabla 49. Matriz Criterios vs Opciones	40
Tabla 50. Características para opciones de cámara de video	42
Tabla 51. Criterios y Opciones.....	42
Tabla 52. Criterios de Priorización	42
Tabla 53. Matriz Precio	43
Tabla 54. Matriz Compatibilidad De Memoria Externa	43
Tabla 55. Matriz Compatibilidad De Cámaras.....	43
Tabla 56. Matriz Transmisión En Tiempo Real	43
Tabla 57. Matriz Criterios vs Opciones	43
Tabla 58. Características para opciones de programación del código	45
Tabla 59. Criterios y Opciones.....	45
Tabla 60. Criterios de Priorización	46
Tabla 61. Matriz Compatibilidad	46
Tabla 62. Matriz Tiempo de Compilación	46
Tabla 63. Matriz Plugin Adicional	46
Tabla 64. Matriz Precio	46
Tabla 65. Matriz Criterios vs Opciones	47
Tabla 66. Prueba general de comunicación para placas ESP82-66 y ESP-32	63
Tabla 67. Tiempos de respuestas placa ESP con HTTP, MySQL, get/post, Arduino IDE.....	64
Tabla 68. Prueba de funcionamiento para placa ESP con MySQL, PHP, get/post y Arduino IDE	65
Tabla 69. Tiempos de respuestas placa ESP con WebSockets, PHP, Arduino IDE	66
Tabla 70. Prueba de funcionamiento para placa ESP con WebSockets, PHP y Arduino IDE.	66
Tabla 71. Tiempos de respuestas placa ESP con MySQL, Arduino IDE	68
Tabla 72. Prueba para funcionamiento de placa ESP con MySQL y Arduino IDE.....	68
Tabla 73. Tiempos de respuestas placa ESP con MQTT-SN, Arduino IDE.....	69
Tabla 74. Prueba de funcionamiento para placa ESP con MQTT-SN y Arduino IDE	70
Tabla 75. Rúbrica de evaluación	72
Tabla 76. Cuadro comparativo entre pruebas realizadas.....	73
Tabla 77. Evaluación del producto final	74
Tabla 78. Comparación de resultados para consumo energético durante 15 días.....	84
Tabla 79. Comparación de resultados para consumo energético durante 1 año.	85

Índice de Figuras

Figura 1. Esquema de circuito que se compone el sensor PIR.....	9
Figura 2. Esquema de elementos dispuestos en el prototipo experimental.....	14
Figura 3. Placa ESP-01 y sus componentes	15
Figura 4. Placa Wemos Mini D1	16
Figura 5. Placa ESP8266 NodeMcu v1.0 / V3 y sus componentes.....	16
Figura 6. Placa ESP-WROVER-KIT-VB, parte frontal y sus componentes	21
Figura 7. Placa ESP32-LyraT y sus componentes	22
Figura 8. Placa ESP32-DevKitC y sus componentes.....	23
Figura 9. Sensor Microondas HB100, vista frontal y posterior	26
Figura 10. Sensor ultrasónico que detecta por medio de un sonar.....	26
Figura 11. Sensor para detectar temperatura y/o humedad DHT11	29
Figura 12. Sensor para detectar temperatura y/o humedad DHT22	30
Figura 13. Sensor para detectar gas MQ-2.....	33
Figura 14. Sensor para detectar gas MQ-5	33
Figura 15. Sensor para gases inflamables MQ-6.....	34
Figura 16. Motor de 5 voltios junto a su chip para secuencia de pasos	37
Figura 17. Motor DC y cableado.....	37
Figura 18. Servomotor SG 90	38
Figura 19. Placa ESP32-CAM, vista frontal y trasera.....	41
Figura 20. ESP-EYE, vista frontal y trasera.....	41
Figura 21. Declaración e inicialización de variables generales.	48
Figura 22. Comunicación con el servidor para interactuar con los dispositivos utilizados.	49
Figura 23. Código para reconectar en caso de perder la comunicación entre el servidor y cliente.	49
Figura 24. Configuración de sensores y actuadores de acuerdo al pin utilizado.	50
Figura 25. Configuración de variables para emparejar el cliente con el servidor.....	50
Figura 26. Librerías para conexión inalámbrica Wi-Fi para tarjetas de desarrollos ESP	51
Figura 27. Librería para suscribirse a uno o varios topics como cliente MQTT-SN.	51
Figura 28. Librería de sensores de temperatura DHT para tarjetas ESP8266 y ESP32.....	51
Figura 29. Librerías para cargar y ejecutar la placa ESP32-CAM.....	52
Figura 30. Diagrama del circuito para el módulo dormitorio uno y dos con sus respectivos actuadores.....	53
Figura 31. Código para confirmar si un foco está encendido o no.....	53
Figura 32. Código para rotar el motor.....	54
Figura 33. Código de secuencia en ocho pasos para mover y detener las bobinas del motor..	54
Figura 34. Diagrama del circuito para el módulo comedor y cocina con sus respectivos sensores y actuadores	55
Figura 35. Código para obtención de datos en el sensor de gas/humo MQ - 6.....	55
Figura 36. Diagrama del circuito para el módulo sala - pasillo con sus sensores y actuadores.	56
Figura 37. Código para medir temperatura con sensor DHT11 cada cinco segundos.	57
Figura 38. Diagrama del circuito para placa ESP32-CAM.....	57
Figura 39. Método para inicializar la transmisión de imágenes y video en tiempo real.	58
Figura 40. Procedimiento para capturar y enviar imágenes a Telegram	58

Figura 41. Diagrama del circuito para el módulo garaje - baño con sus respectivos sensores y actuadores.....	59
Figura 42. Código para detectar movimiento en el sensor de presencia HC-SR501	60
Figura 43. Diagrama lógico del circuito completo en bloques para hardware.....	61
Figura 44. Diagrama físico del circuito completo en bloques para hardware.....	62
Figura 45. Contiene el resultado de la tabla 68.	65
Figura 46. Contiene el resultado de la tabla 70.	67
Figura 47. Contiene el resultado de la tabla 72.	69
Figura 48. Contiene el resultado de la tabla 74.	71
Figura 49. Protocolos domóticos utilizados para la comunicación especificando su velocidad de transmisión, distancia de alcance y medio de transmisión.	72
Figura 50. Representación en barras para comparación de protocolos de comunicación.	74
Figura 51. Función publish () para envío de datos.	76
Figura 52. Método callback () para descifrar datos recibidos en el cliente.	76
Figura 53. Fórmula para calcular consumo de energía eléctrica para un foco	79
Figura 54. Pruebas ejecutadas en un mismo escenario durante 30 días.	79
Figura 55. Variables para cálculo de consumo de energía eléctrica en prueba A.	80
Figura 56. Cálculo de consumo diario de energía para un foco	80
Figura 57. Cálculo de consumo mensual de energía para un foco	81
Figura 58. Cálculo de consumo anual de energía para un foco.....	81
Figura 59. Consumo promedio de 6 horas diarias en dólares sin intervención del grupo de trabajo.....	81
Figura 60. Ecuación para calcular precio unitario en un día.	82
Figura 61. Variables para cálculo de consumo de energía eléctrica en prueba B.	83
Figura 62. Cálculo de consumo diario de energía para un foco	83
Figura 63. Cálculo de consumo mensual de energía para un foco	83
Figura 64. Cálculo de consumo anual de energía para un foco.....	83
Figura 65. Consumo promedio de 3,4 horas diarias en dólares con intervención del grupo de trabajo.....	84
Figura 66. Vista externa del prototipo experimental con un sistema domótico.	85
Figura 67. Vista lateral del prototipo experimental con un sistema domótico.....	86
Figura 68. Vista interna del prototipo experimental con un sistema domótico.....	86
Figura 69. Cableado implementado en el prototipo experimental con un sistema domótico. .	87
Figura 70. Envío de orden para encender focos desde la aplicación web con protocolo MQTT para su comunicación.	88
Figura 71. Encendido de focos en el prototipo experimental con un sistema domótico.....	89
Figura 72. Visualización para cierre de persiana en la aplicación web.....	90
Figura 73. Resultado de orden en persiana desde el cliente hardware.	90
Figura 74. Apertura de persiana en el prototipo experimental.	91
Figura 75. Transmisión en tiempo real desde ESP32-CAM.	91
Figura 76. Alerta recibida en el dispositivo móvil del usuario desde ESP32-CAM.....	92
Figura 77. Envío de imágenes en tiempo real desde el cliente hardware ESP32-CAM.	92
Figura 79. Alerta de movimiento no autorizado entregada por el sensor PIR a la interfaz de la aplicación web.....	93
Figura 80. Alerta de movimiento no autorizado en cloud MQTT enviado por el cliente hardware que cuenta con un sensor PIR.....	94
Figura 81. Envío de variación de temperatura desde el sensor DHT-11.....	94

Figura 82. Variación de temperatura entregada por el cliente hardware en interfaz de la aplicación web.....	95
Figura 83. Alerta por detección de gas entregada por el cliente hardware en interfaz de la aplicación web.....	96
Figura 84. Alerta por detección de gas enviada desde el cliente hardware que cuenta con un sensor MQ-6.....	96

Resumen

Hoy en día, la seguridad del hogar es uno de los temas más críticos, es por esta razón, que métodos tradicionales como cercar o elevar muros en las estructuras ya no es suficiente. El propósito de este proyecto técnico fue presentar una opción viable al usuario final, que requiera conocer el estado actual de su domicilio desde cualquier lugar y momento vía Internet, a su vez, un sistema de control domótico que reduzca de alguna manera el consumo energético del mismo. Para esto, se investigó principios relacionados con Domótica, Sistemas Embebidos e Internet de las Cosas, permitiendo el diseño, desarrollo e implementación del prototipo.

De este modo, tarjetas de desarrollo, sensores y actuadores empleados fueron analizados y comparados por medio de métodos de priorización, técnica recomendada para la toma de decisiones; por consiguiente, se integraron los elementos mencionados entre sí de acuerdo a los requisitos planteados, diseñando bloques de hardware con comunicación inalámbrica que posteriormente fueron implementados en un prototipo experimental. Al comparar evidencias entre los protocolos HTTP, WebSockets, MySQL y MQTT, se demostró que MQTT fue ideal con un 96% de efectividad para establecer una respuesta inmediata a cada petición del usuario, además de ser el único que cuenta directamente con “QoS”.

Finalmente, una vez implementado y verificado el prototipo, se destaca su autonomía energética puesto que anualmente reduce el consumo de luz en un 43,33 %, además, incorpora ventajas al momento de prevenir al usuario en caso de una intrusión gracias a su disponibilidad 24/7.

Abstract

Nowadays, home safety is one of the most critical issues in society, which is why traditional methods such as fencing or raising walls in structures are no longer enough. The purpose of this technical project was to present a viable option to the end users who may require to know the current state of their home from anywhere and at any time via the Internet. In addition to this, a home domotic control system that somehow reduces energy consumption was proposed. In order to develop this proposal, principles related to Domotics, Embedded Systems and Internet of Things were investigated to foster the design, development and implementation of the product.

Thereby, development boards, sensors and employed actuators were analyzed and compared by means of prioritization methods which is a recommended technique for decision making in these kinds of procedures. These elements became integrated according to the intended requirements, designing hardware blocks with wireless communication that were later implemented in an experimental prototype. By comparing evidence between HTTP, WebSockets, MySQL and MQTT protocols, it was demonstrated that MQTT was ideal with 96% effectiveness to establish an immediate response to each user request, besides being the only one that has "QoS" specifically.

Finally, once the prototype has been implemented and verified, its energy autonomy stands out because it reduces light consumption by 43,33 % per year; furthermore, it incorporates advantages at the moment of preventing the user in case of an intrusion thanks to its 24/7 availability.

Introducción

Según Grupo EL COMERCIO (2020), en el año 2019 en Quito, la inseguridad se incrementó en un 17%. Por esta razón, los sistemas caseros y/o tradicionales no son suficientes para asegurar el domicilio, por otra parte, Moreno Vargas & Valdez Bravo (2015) afirman que: el 52% de un grupo de personas encuestadas desearían contar con un sistema de video seguridad en el hogar (pág. 82). En el mismo año, en Ecuador, el consumo energético se incrementó en un 4,5% (Ministerio de Energía y Recursos Naturales no Renovables, 2020) generando un aumento en el pago de la planilla de luz; dicho incremento se realizó por el uso incorrecto de focos, electrodomésticos, entre otros.

Con el aumento de la tecnología se sabe que, el 36% de hogares ecuatorianos pueden acceder a Internet (Instituto Nacional de Estadística y Censos (INEC), 2016, pág. 8), de manera que los usuarios pueden comunicarse de forma inalámbrica (WiFi) simplificando y automatizando servicios.

Por esta razón el proyecto técnico será orientado a la necesidad de incrementar la seguridad en hogares mediante la elaboración de un diseño y ejecución con control domótico, dispuesta en un prototipo experimental, utilizando la herramienta Arduino IDE para ejecutar el envío y recepción de datos; permitiendo a los usuarios obtener seguridad, confort y ahorro energético.

Problema

Con el aumento de la tecnología en los hogares se tiene al menos uno o más dispositivos tecnológicos con conexión a Internet, esto conlleva a que muchas de las personas que habitan en dichos hogares desconozcan sobre la importancia de tener un control adecuado de cada dispositivo.

Al no tener un correcto control de los dispositivos tecnológicos con conexión a internet dentro de los hogares, el consumo de energía aumenta considerablemente, esto depende de la cantidad de dispositivos que estén conectados, los mismos que consumen energía y se ve reflejado en

las planillas de luz con costos muy altos. Debido a este avance continuo se demuestra la necesidad de instalaciones inteligentes permitiéndonos presentar un diseño e implementación con la capacidad de incluir o excluir módulos a ser automatizados.

Y el otro gran problema existente se centra específicamente en la inseguridad actual de nuestro país debido a que, si por algún motivo el domicilio se muestra abandonado, estamos expuestos a ser víctimas de invasión a la propiedad privada.

Justificación

En la ciudad de Quito se evidencia la inseguridad y consumo excesivo de energía eléctrica en propiedades privadas, por lo que es necesario implementar sistemas automáticos con control domótico que permitan tener accesibilidad y disponibilidad con motivo de seguridad y/o ahorro energético para así concientizar al usuario final.

Es importante destacar que se realiza el diseño e implementación con un prototipo didáctico que posea módulos o sensores para un escenario cambiante, es decir, que el usuario final tenga la potestad de agregar o eliminar módulos a ser automatizados.

Además, la importancia de diseñar la aplicación web es demostrar un correcto control del hardware que conforma un sistema domótico, ya que permitirá al usuario configurar diferentes dispositivos, que realicen acciones por medio de dispositivos tecnológicos.

Con el desarrollo de la aplicación web se reducirá el consumo innecesario de energía eléctrica ya que el usuario tendrá un control adecuado de los dispositivos eléctricos y tecnológicos que se encuentren dentro del hogar de esta forma se reducirá el pago de planillas eléctricas elevadas.

Mediante la implementación del sistema domótico (hardware y software) se mejorará dentro del hogar ya que el sistema enviará y recibirá datos en tiempo real.

Objetivos

Objetivo general

Diseñar e implementar un sistema de control domótico (con Sistemas Embebidos) en un prototipo experimental para conectarse vía Internet con una aplicación web basada en IoT.

Objetivos específicos

Investigar la información, el funcionamiento, la conexión de los distintos dispositivos y posibles herramientas, que se manipulan en un sistema de control domótico con sistemas embebidos.

Analizar la información recopilada y seleccionar herramientas de hardware que se utilizan en un sistema de control domótico con sistemas embebidos.

Analizar los datos que se generan a través de sensores y actuadores en un sistema de control domótico con sistemas embebidos.

Diseñar la metodología de implementación en un sistema de control domótico para modelar el prototipo experimental.

Desarrollar la metodología de implementación en base a los requerimientos del análisis inicial.

Implementar el sistema de control domótico con sistemas embebidos en un prototipo experimental.

Ejecutar las pruebas de funcionamiento para comprobar que los datos enviados y recibidos son confiables.

Optimizar el uso de elementos y/o dispositivos en un determinado ambiente, incrementando el confort de los usuarios que tienen acceso a este sistema, controlado por medio de domótica con sistemas embebidos en un ambiente domiciliario.

Marco metodológico

Para el desarrollo del prototipo se representará gráfica con varios módulos distribuidos de forma estratégica que permita su comunicación de forma inalámbrica, mismos que enviarán y

recibirán datos alojados en tiempo real en una base de datos con un servidor externo que podrán ser visualizados en una página web.

Se tomará como referencia optar por el Modelo en V del Ciclo de Vida, conocido también como modelo de cuatro niveles. Al ser un proyecto técnico que se divide en dos módulos que son Hardware y Software, esta metodología es apropiada para poder realizar el desarrollo y pruebas mediante varias etapas específicamente para hardware:

Fase 1 - Definición de especificaciones: Se definirá y documentará los diferentes requisitos del sistema a desarrollar, identificando los valores numéricos más concretos posibles (Pérez, Berreteaga, Ruiz de Olano, Urkidi, & Perez, 2006).

Fase 2 - Diseño global: Se obtendrá un diseño y visión general del prototipo experimental. (Pérez et al., 2006)

Fase 3 - Diseño en detalle: Se detallará cada bloque de manera minuciosa para la fase dos. (Pérez et al., 2006)

Fase 4 - Implementación: Se materializará el diseño en detalle. (Pérez et al., 2006)

Fase 5 - Test unitario: Se verificará cada módulo hardware de forma unitaria, comprobando su funcionamiento adecuado. (Pérez et al., 2006)

Fase 6 - Integración: Se integrarán los distintos módulos que forman el sistema del prototipo. De igual manera, se generará un documento que registre las pruebas tanto de la fase cinco como de la fase seis. Por último, se comprobará el cumplimiento de los requisitos establecidos. (Pérez et al., 2006)

Fase 7 - Test operacional del sistema: Se realizarán las pruebas finales sobre un escenario final, registrando una vez más las pruebas realizadas y los resultados obtenidos. (Pérez et al., 2006)

CAPÍTULO 1

1. Marco Teórico

En este capítulo se puntualizará la información relacionada con los elementos físicos y lógicos utilizados a lo largo del desarrollo de este tema. Además, se incluirán definiciones puntuales posteriores a una investigación que categoriza desde información global hasta una descripción específica que exige este proyecto técnico para su diseño e implementación.

1.1. Internet de las Cosas (IoT)

El actual desafío para los sistemas de información se centraliza en una manera eficaz e invisible para almacenar, procesar y presentar cantidades inmensas de datos producidas por redes de sensores e identificación por radiofrecuencia (RFID) incluidos en productos para el usuario final, siendo la conectividad inteligente de estas redes una parte indispensable (Gubbi, Marusi, Buyya, & Palaniswam, 2013).

1.2. Sistemas embebidos

Un Sistema Embebido es un sistema de propósito específico en el cual, la computadora es contenida parcial o completamente por el dispositivo que lo controla. A diferencia de los computadores de propósito general, los sistemas embebidos son diseñados para una aplicación específica, es decir, estos sistemas son programados previamente a su implementación, de manera que, una vez diseñado el sistema, se torna complicado alterar su funcionalidad (Camargo, 2011, pág. 27).

Entonces, se cuenta con un nivel de autonomía y celeridad en procesos llevados a cabo, permitiendo el uso de placas desarrolladas con sistemas embebidos con ventajas como un precio accesible comparado con un sistema de control producido por industrias internacionales. En conclusión, se puede sacar el máximo provecho a estas ventajas en capacidad y flexibilidad trabajando en proyectos que permitan automatizar procesos y servicios en el hogar, simplificando temas de seguridad y confort para las personas (Rosero, 2016, pág. 22).

1.3. Domótica

Podría decirse que “Se define Domótica a las agrupaciones automatizadas de equipos, que pueden comunicarse interactivamente entre ellos integrándolos a través de un bus multimedia” (Junestrand, Passaret, & Vázquez, 2004, pág. 4). Actualmente se considera a la automatización de hogares como mercado potencial por los siguiente:

1.3.1. Viviendas existentes.

Es de conocimiento público que el parque de viviendas es un mercado de interés, debido a las exigencias actuales del usuario final que busca confort, seguridad y ahorro energético tanto en viviendas existentes como en viviendas de nueva construcción. Además, el coste para adquirir estos sistemas es accesible y mínimo en comparación a los beneficios garantizados (Medina Valle, 2012, pp. 14-16).

1.3.2. Preservación de la naturaleza y ahorro energético.

Actualmente una de las misiones más importantes del ser humano es cuidar y conservar el medio ambiente, pues las investigaciones ambientales a nivel mundial demuestran que, si no existe una concientización sobre el uso de los recursos del planeta, estamos sujetos al agotamiento de los mismos. Es así como la Domótica tiene como fin ser un medio de energía verde, es decir, una participación directa sobre la manera en cómo se administran los recursos (Medina Valle, 2012, pp. 14-16).

1.3.3.Desarrollo de comunicaciones.

Este mundo de comunicaciones vía Internet, así como sus servicios presentan una evolución constante. Gracias a estos avances, el factor de automatización es viable dentro de los hogares, principalmente para tener un control automático sobre temas simples como activar o apagar una luz, o bloquear y desbloquear una puerta por motivos de seguridad mediante una red de cableado o red inalámbrica (Medina Valle, 2012, pp. 14-16).

1.4. Arduino

Es de importancia mencionar que Arduino es el hardware de código abierto (OSHW) con mayor impacto en este último tiempo debido a su simplicidad para uso y modificación, además cuenta con una comunidad que aporta constantemente maneras innovadoras de utilizar este hardware desde prototipos a escala hasta productos finales relacionados con la ingeniería como es el caso de la automatización de hogares, siendo este el tema de principal importancia para este proyecto técnico.

1.5. Hardware

Se puntualizará la información relacionada con los elementos físicos utilizados para este proyecto técnico.

1.5.1. Tarjeta de desarrollo ESP-8266.

La placa ESP-8266 aparece como una solución de SoC Wi-Fi, es decir, que el sistema está incorporado en un chip, permitiendo una comunicación inalámbrica con transmisión de datos en tiempo real, ofreciendo una mejor experiencia al usuario final que requiere contenido multimedia en sus productos, así como el “uso eficiente de energía debido a su diseño compacto y rendimiento garantizado en la industria IoT” (Espressif Inc, 2019d, pág. 1).

Siendo un módulo autónomo, puede ser utilizado como servidor o cliente para diferentes tipos de aplicaciones, además de poseer una *“alta velocidad, incrementa el rendimiento del sistema y optimiza su memoria”*. Así mismo, cuenta con pines digitales (GPIO) destinados para conectar sensores empleados en la automatización de hogares (Domótica) (Espressif Inc, 2019d, pág. 1).

1.5.2. Tarjeta de desarrollo ESP-32 DevKit3.

La placa ESP32 es utilizada para una gama extensa de aplicaciones gracias a la integración de Wi-Fi, Bluetooth y Bluetooth Low Energy, permitiendo que el módulo genérico sea muy potente. Podrá ser puesto en práctica desde una trama de sensores de potencia mínima hasta

adaptaciones “más exigentes como transmisión y decodificación MP3, p codificación para la voz” (Espressif Inc, 2019c, pág. 1).

1.5.3.Cámara de video ESP32 CAM.

La cámara ESP32-CAM es un dispositivo que se compone de dos módulos, uno es la placa principal y el otro es una pequeña cámara que, tiene la capacidad de funcionar independientemente pues dispone de conexión inalámbrica (WiFi), además de ser utilizado como una solución viable para aplicaciones de IoT (Brandão, 2019).

1.6.Módulos

1.6.1.Sensores.

1.6.1.1. Red de sensores (WSN).

En la actualidad es una de las tecnologías con mayor auge debido a su principal objetivo, que es la posibilidad de implementar equipos con acceso a Internet de manera inalámbrica en zonas que no poseen infraestructura. Esto presenta una notable ventaja facultando un monitoreo en tiempo real para condiciones como la humedad, temperatura o luz, así como el envío de alertas por fuga de gas, presencia de humo o el ingreso de personas no autorizadas a la vivienda en caso de existirlas. En otras palabras, una red “WSN” posee la capacidad de enviar y recibir datos de forma inalámbrica.

Sin embargo, al ser un factor esencial para poder realizar proyectos de Domótica, es necesario prestar atención a sus vulnerabilidades al momento de establecer una comunicación vía Wi-Fi.

1.6.1.2. Sensor detector de movimiento pasivo infrarrojo HC-SR501 (PIR).

La Figura 1, manifiesta a un dispositivo que detecta movimiento en un rango de 7 metros, además, cuenta con un precio accesible y no se deteriora. Sus funciones integradas son:

- Parámetros modificables: por medio de 2 potenciómetros, se tiene la potestad de alterar su sensibilidad y distancia para detectar movimiento (Punto Flotante S.A., 2017, pág. 3).

- Detección automática de luz: mediante una fotorresistencia de Sulfuro de Cadmio (CdS), se deshabilita la actividad del módulo en caso de existir luz visible en el rango de detección (Punto Flotante S.A., 2017, pág. 3).

Sensor Detector de Movimiento Pasivo Infrarrojo HC-SR501 (PIR)



Figura 1. Esquema de circuito que se compone el sensor PIR
Fuente: (Punto Flotante S.A., 2017, pág. 5)

1.6.1.3. Sensor de temperatura/humedad DHT11.

Según Mouser Electronics (2019), “el sensor de temperatura/humedad DHT11 cuenta con complejos sensores con salida de señal digital calibrada; a su vez, incluye un componente para medir humedad del tipo resistivo y un componente para medir temperatura NTC conectado a un microcontrolador con 8 [bit] de alto rendimiento, ofreciendo calidad, respuesta rápida, capacidad anti interferencia y rentabilidad” (Mouser Electronics, 2019, pág. 2).

1.6.1.4. Sensor de gas y humo MQ6.

El sensor MQ6 detecta gases inflamables como el gas propano (GLP), pues posee SnO₂ como material sensible, es decir, posee mínima conductividad en aire limpio. Por ese motivo cuando existe gas inflamable en el medio ambiente, la conductividad del sensor se incrementa. Además, el sensor cuenta con sensibilidad de amplio rango, vida útil prolongada y una mínima inversión económica a cambio de beneficios relacionados con la seguridad y bienestar del hogar (Zhengzhou Winsen Electronics Technology CO., LTD, s. f.).

1.6.1.5. LDR.

Una resistencia dependiente de luz o fotorresistencia se define como “una célula de Sulfuro de Cadmio, envuelta con una resina cristalina que resiste la humedad, además, su respuesta espectral es similar a la del ojo humano” (CEBEK, s. f.) Es decir, es un elemento que varía dependiendo de la intensidad de luz para principalmente detectar luminosidad.

1.6.2. Actuadores.

Se conoce como actuadores a los elementos que tienen la capacidad de cambiar su estado mediante un proceso de recibir órdenes por intermedio de una interfaz, un regulador o controlador. En este proyecto técnico se utiliza Arduino IDE para generar dichas órdenes.

1.6.2.1. Diodo emisor de luz (LED).

Es una fuente de luz semiconductor de dos diferentes conductores, siendo uno del tipo P y otro del tipo N. Al aplicar un voltaje de avance suficiente hará que los orificios (conductor tipo P) y los electrones (conductor tipo N) “se recombinen en la unión P-N, liberando energía en forma de luz” (ROHM CO., LTD., s. f.).

1.6.2.2. Motor modelo 28BYJ-48 paso a paso.

También conocido como “stepper” es un dispositivo electromagnético que trabaja con una conversión de energía eléctrica a energía mecánica de rotación. Su principal característica es que hacen el movimiento de “un giro elemental del rotor” conocido como un paso por cada impulso que recibe (Fernández Aragón, 2011, pág. 5).

Además, “este motor practica el principio físico equivalente al de motores eléctricos rotatorios de corriente continua y alterna” (Germán Corona Ramírez et al., 2014, pp. 284). Sin embargo, para este caso particular, la conmutación se logra a través de algún sistema electrónico o tarjeta microcontroladora, como es el caso de la tarjeta Arduino® para este proyecto técnico (Germán Corona Ramírez et al., 2014, pp. 284-285)

1.6.2.3. Módulo integrado para motor ULN2003.

Conocido también como un arreglo de transistores Darlington. Se utiliza este integrado para realizar la conexión a otras placas programables como Arduino o ESP32 entre las más utilizadas.

1.7. Software

Se puntualiza la información relacionada con los componentes lógicos utilizados para este proyecto técnico.

1.7.1. Arduino IDE.

Arduino IDE es un ambiente de desarrollo integrado empleado para editar, compilar y depurar código mediante su interfaz gráfica (GUI), por este motivo, “esta interfaz permite desarrollar códigos, contiene un área de mensajes, herramientas con acceso a funciones comunes y una disposición de menús” (Arduino, 2015).

Esta será la principal herramienta a utilizarse en este proyecto técnico para programar los dispositivos mencionados con anterioridad.

1.7.2. PHP.

Hypertext Preprocessor permite al usuario elaborar órdenes adecuadas para desarrollo web, mismo que puede ser introducido en HTML” (PHP Group, s. f.).

Su ventaja es la simplicidad que presenta en su estructura tanto en el desarrollo de scripts para servidor como para clientes. Además, existe compatibilidad con el hardware y software de las tarjetas de desarrollo utilizadas en este sistema de control domótico.

1.7.3. Message queue telemetry transport (MQTT).

Es un protocolo para mensajería que tiene como objetivo vincular dispositivos embebidos y redes mediante aplicaciones que permiten al usuario interactuar con páginas web. Además, “MQTT está construido en el tope del protocolo TCP, y entrega mensajes mediante tres niveles de calidad de servicio (QoS)” (Ramírez & Pedraza, 2017).

Así mismo, se sabe que MQTT está compuesto de tres entidades: subscriber, publisher y broker. En primer lugar, el dispositivo subscriber recibirá información del broker en topics específicos de los demás publishers, y posteriormente el publisher transmitirá la información a través del broker a los subscribers de interés. “Por lo tanto, el protocolo MQTT representa un protocolo de mensajería ideal para las comunicaciones IoT y M2M que posean dispositivos pequeños, económicos, de baja energía, y de poca memoria” (Ramírez & Pedraza, 2017).

1.7.4. MQTT – SN.

MQTT para redes de sensores está elaborado para alinearse de forma similar a MQTT, con la diferencia que está adaptado para un entorno con comunicación inalámbrica como: “poco ancho de banda, altas fallas de enlace, longitud corta para los mensajes, entre otros”; además, es un protocolo optimizado para aplicaciones con dispositivos de bajo costo alimentados con poco voltaje o baterías, capacidad restringida de procesamiento y/o almacenamiento. (Stanford-Clark & Linh Truong, 2013, pág. 4).

CAPÍTULO 2

2. Análisis Hardware - Software

En este capítulo se analizará y comparará información relacionada con los elementos físicos y lógicos, donde se incluirán características específicas de cada uno de ellos con el fin de demostrar los criterios de selección dependiendo de los equipos optados, que dan lugar a la ejecución del prototipo experimental.

2.1. Análisis de Hardware

Para esta sección se considerarán criterios generalmente utilizados para seleccionar un elemento electrónico basándonos en el precio, disponibilidad, compatibilidad de tecnologías, capacidad de procesamiento, cobertura, nivel de respuesta, entre otros. Estos criterios serán evaluados de acuerdo a la técnica de priorización aplicada en base al aprendizaje adquirido durante la carrera universitaria que ayudará al equipo de trabajo en la toma de decisiones.

2.1.1. Representación gráfica del prototipo experimental.

La Figura 2, representará gráficamente el prototipo experimental que será utilizado para su implementación. Se dispondrá de varias placas de desarrollo ESP ubicadas estratégicamente para obtener una comunicación inalámbrica estable, a las mismas que se integran sensores y actuadores dependiendo de su posición y función.

Representación gráfica del prototipo experimental

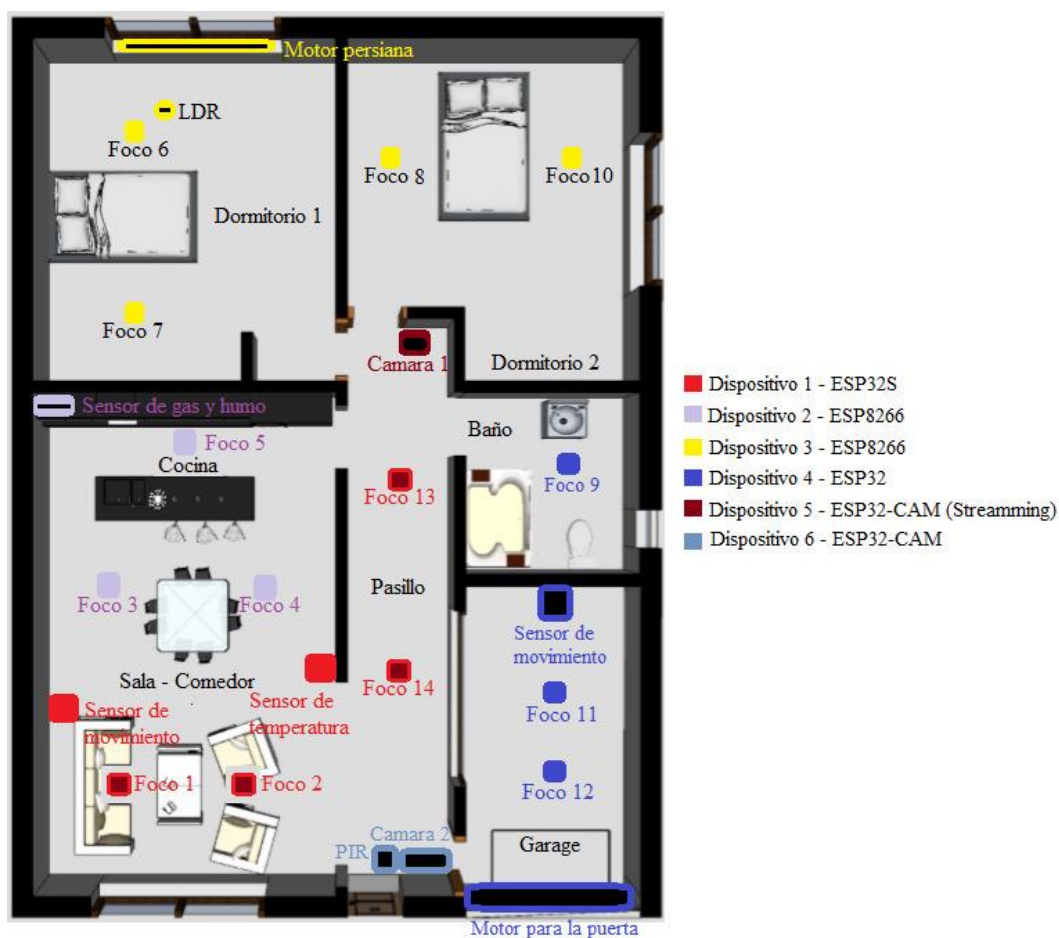


Figura 2. Esquema de elementos dispuestos en el prototipo experimental
Elaborado por: Juan Fonseca y Daniel Soria

2.1.2. Análisis tarjeta de desarrollo ESP-8266.

Se detallarán las versiones más representativas para esta tarjeta de desarrollo. También se destaca que son placas baratas y fáciles de programar para diferentes usos, entre ellos la domótica.

2.1.2.1. ESP-01.

La placa WiFi ESP-01 mostrada en la Figura 3, posee un procesador central ESP8266, el cual es líder en la industria de ultra baja potencia. Entre sus principales características se destaca su soporte de velocidad de reloj para 80 MHz y 160 MHz, soporta RTOS (Sistema Operativo

de Tiempo Real), Wi-Fi integrado MAC/BB/RF/PA/LNA con antena incorporada (AI-Thinker, 2015, pág. 3).

Además, soporta el estándar IEEE802.11 b/g/n junto a la pila de protocolos TCP/IP, dando a los usuarios la posibilidad de agregar los módulos a una red de dispositivos existentes, o a la construcción de un controlador de red independiente (AI-Thinker, 2015, pág. 4).

Placa de desarrollo ESP-01

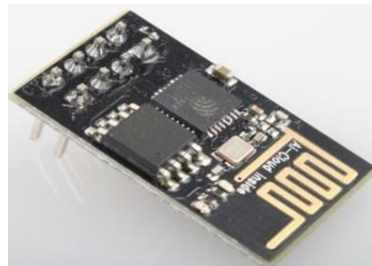


Figura 3. Placa ESP-01 y sus componentes
Fuente: (AI-Thinker Team, 2015, pág. 10)

2.1.2.2. Wemos mini D1.

Wemos Mini D1 es una placa diminuta Wi-Fi basado en el chip ESP8266EX, siendo una solución muy compacta para el diseño de prototipos pequeños e inteligentes conectados a la World Wide Web gracias a las funcionalidades Wi-Fi del Espressif ESP8266. Cuenta con 4MB de memoria flash, 80MHz de reloj del sistema, alrededor de 50k de RAM utilizable y un transceptor Wi-Fi, entre otros (Zerynth, s. f.).

Por lo tanto, las ventajas notables de esta placa son su puerto Micro SD, un conversor serial y un regulador de tensión con alimentación de 5V; sin embargo, su mayor desventaja se encuentra en el menor número de pines GPIO disponibles para su uso (11) comparado con la NodeMCU que cuenta con 17 pines GPIO (Llamas, 2018). La Figura 4, muestra el diseño frontal y posterior de esta placa.

Wemos Mini D1

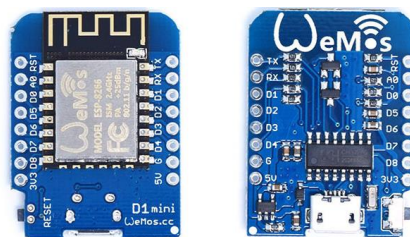


Figura 4. Placa Wemos Mini D1

Fuente: (Zerynth, s. f.)

2.1.2.3. NodeMCU v1.0 / V3.

La tarjeta de desarrollo NodeMCU representada en la Figura 5, es la más popular entre todas sus versiones porque es de conocimiento general que es una placa autónoma, pues tiene incorporado una antena WiFi, un procesador potente, un conector USB que permite conectar directamente a la computadora y trabajar, entre otros. Además, cuenta con bastante documentación y librerías proporcionadas por la “Comunidad Arduino” que día a día contribuye con soluciones, guías, código fuente, entre otros y posee compatibilidad con Arduino IDE, indispensable para desarrollar software relacionado con Domótica e IoT (Handson Technology, s. f.).

ESP8266 NodeMCU

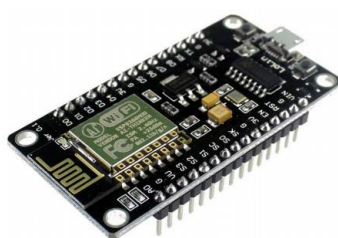


Figura 5. Placa ESP8266 NodeMcu v1.0 / V3 y sus componentes.

Fuente: (Handson Technology, s. f.)

2.1.2.4. Análisis de priorización para placas ESP-8266.

En esta sección se presentarán las características (mostradas en la Tabla 1) de los elementos investigados y relacionados con las placas ESP-8266 y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 1. Características para opciones de placa ESP-8266

Parámetro	ESP-01	WeMos D1 mini	NodeMCU
Microcontrolador	ATmega328P	ESP-8266EX	ESP-8266EX
CPU	Tensilica Xtensa LX3 32-bit	Tensilica 32-bit RISC CPU Xtensa LX106	Tensilica 32-bit RISC CPU Xtensa LX106
Memoria Flash	1 MB	4 MB	4 MB
Velocidad Reloj	80 [MHz]	80 - 160 [MHz]	80 - 160 [MHz]
Señal de salida	Digital	Analógica / Digital	Analógica / Digital
Alimentación	3,0 - 3,6 [V]	3,3 [V]	3,3 - 5 [V]
Conector micro USB	-	Tipo B	Tipo B
Protocolo WiFi	802,11 b/g/n	802,11 b/g/n	802,11 b/g/n
Protocolos de Seguridad Inalámbrica	WPA/WPA2	WEP WPA/WPA2	WEP/TKIP/AES WPA/WPA2
Número de pines	8	12	18
Precio	4,5 USD	10 USD	7,5 USD
Dimensiones	14,3*24,8*3,0 [mm]	34,2*25,6 [mm]	58*31*13 [mm]

Nota: Esta tabla contiene las características principales de las placas analizadas para realizar la priorización de la placa ESP-8266.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 2, facilitará el análisis de matrices posteriores mediante el uso de valores numéricos en la calificación de placas, sensores y actuadores.

Tabla 2. Ponderación de criterios

RESPUESTAS	VALOR
Igualmente importante	1
Significativamente importante	5
Excesivamente importante	10
Significativamente menos importante	1/5
Excesivamente menos importante	1/10

Nota: Esta tabla contiene los valores con los que se calificará y concluirá la opción recomendada para todos los análisis a realizarse en este capítulo.

Elaborado por: Juan Fonseca y Daniel Soria

Para realizar un análisis de priorización será necesario la Tabla 3, misma que contiene criterios de precio, capacidad de procesamiento, nivel de respuesta y disponibilidad; estos serán argumentados mediante el uso de la Tabla 1, misma que contiene detalles específicos de cada producto.

Tabla 3. Criterios y Opciones

CRITERIOS	OPCIONES
PRECIO	ESP-01 WEMOS MINI D1 NODEMCU V1.0 / V3
CAPACIDAD DE PROCESAMIENTO	
NIVEL DE RESPUESTA	
DISPONIBILIDAD	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 4. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Precio	Capacidad Procesamiento	Nivel de respuesta	Disponibilidad	Valor	Ponderado definido
1	Precio		10	1	5	16	0.604
2	Capacidad procesamiento	1/10		5	1	6,1	0,230
3	Nivel de respuesta	1	1/5		1	2,2	0,083
4	Disponibilidad	1/5	1	1		2,2	0,083
					TOTAL	26,5	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para la placa ESP-8266.
Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 4, presentará una matriz de comparación entre fila y columna, donde se comprobará el criterio que tiene más relevancia; para ello se deberá preguntar si el criterio de la fila es más importante que el criterio de la columna. Este proceso se comparará de la siguiente manera:

- Fila 1 contra columnas C, D, E.
- Fila 2 contra columnas B, D, E.
- Fila 3 contra columnas B, C, D.

El puntaje de cada celda comparada será asignado en base a la Tabla 2, teniendo en cuenta que, si su calificación es un valor entero su inverso será un valor fraccionario. Este procedimiento se repetirá en cada una de las celdas.

Para completar las celdas de la columna “Valor”, se realizará una adición a cada una de las filas. En consecuencia, se obtendrá el total mediante la sumatoria de dicha columna. Así mismo, los valores pertenecientes a la columna “Ponderado definido” serán el resultante de la división entre las celdas mencionadas con anterioridad y el “Total”. Finalmente, se sumará la columna “Ponderado definido” donde, si su resultado es uno se concluirá que la matriz es consistente.

Tabla 5. Matriz Precio

#	A	B	C	D	E	F
0	PRECIO	ESP-01	WEMOS D1	NODEMCU	VALOR	Ponderado relativo
1	ESP-01		5	1	6	0,484
2	WEMOS D1	1/5		1/5	0,4	0,032
3	NODEMCU	1	5		6	0,484
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Precio de la placa ESP-8266.
Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 5, analizará entre las opciones de placas propuestas, ¿Cuál es la mejor opción que permitirá desarrollar el código de programación?, tomando en cuenta el criterio “PRECIO”. Para ser comparada, se planteará la pregunta de importancia entre filas y columnas, mismas que serán calificadas haciendo el uso de la Tabla 2; una vez realizado este paso y completada la matriz se proseguirá a completar las celdas de la columna “Valor”, realizando una adición a cada una de las filas. En consecuencia, se obtendrá el total mediante la sumatoria de dicha columna. Así mismo, los valores pertenecientes a la columna “Ponderado relativo” serán el resultante de la división entre las celdas mencionadas con anterioridad y el “Total”.

Finalmente, se sumará la columna “Ponderado relativo” donde, si su resultado es igual a 1 se concluirá que la matriz es consistente.

Tabla 6. Matriz Capacidad de Procesamiento

#	A	B	C	D	E	F
0	CAPACIDAD DE PROCESAMIENTO	ESP-01	WEMOS D1	NODEMCU	VALOR	Ponderado relativo
1	ESP-01		1/5	1/5	0,4	0,032
2	WEMOS D1	5		1	6	0,484
3	NODEMCU	5	1		6	0,484
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Capacidad de Procesamiento de la placa ESP-8266.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 7. Matriz Nivel de Respuesta

#	A	B	C	D	E	F
0	NIVEL DE RESPUESTA	ESP-01	WEMOS D1	NODEMCU	VALOR	Ponderado relativo
1	ESP-01		1/5	1/10	0,3	0,014
2	WEMOS D1	5		1/5	5,2	0,254
3	NODEMCU	10	5		15	0,732
				TOTAL	20,5	1

Nota: Esta tabla compara las opciones con relación al criterio Nivel de Respuesta de la placa ESP-8266.
Elaborado por: Juan Fonseca y Daniel Soria

De la misma manera que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 6, 7 y 8, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 8. Matriz Disponibilidad

#	A	B	C	D	E	F
0	DISPONIBILIDAD	ESP-01	WEMOS D1	NODEMCU	VALOR	Ponderado relativo
1	ESP-01		5	1/5	5,2	0,254
2	WEMOS D1	1/5		1/10	0,3	0,014
3	NODEMCU	5	10		15	0,732
				TOTAL	20,5	1

Nota: Esta tabla compara las opciones con relación al criterio Disponibilidad de la placa ESP-8266.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 9. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	PRECIO	CAPACIDAD DE PROCESAMIENTO	NIVEL DE RESPUESTA	DISPONIBILIDAD	PONDERADO
1	ESP-01	0,604*0,484	0,230*0,032	0,083*0,014	0,083*0,254	0,32194
2	WEMOS MINI D1	0,604*0,032	0,230*0,484	0,083*0,254	0,083*0,014	0,152892
3	NODEMCU	0,604*0,484	0,230*0,484	0,083*0,732	0,083*0,732	0,525168
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción entre las propuestas para la placa ESP-8266,

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 9, obtendrá la relación entre Criterios y Opciones, donde para completarla se realizará lo siguiente:

- Los valores de la columna “Ponderado definido” pertenecientes a la Tabla 4, serán colocados de manera horizontal para cada fila de acuerdo a su criterio.
- Los valores de las columnas “Ponderado relativo” pertenecientes a las Tablas: 5, 6, 7 y 8; serán colocados de manera vertical para cada columna de acuerdo a su opción.
- Los valores presentados en cada celda deberán ser multiplicados.
- Para conseguir el resultado final de la matriz se sumarán los valores de la columna “Ponderado”, mismos que serán el producto de la adición de cada una de las filas.

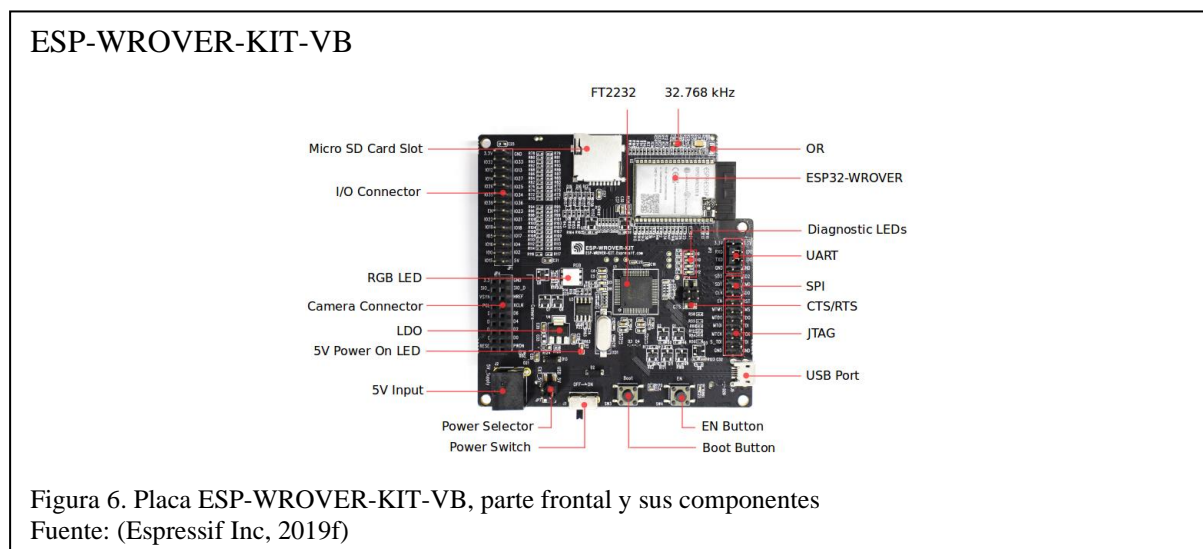
Por consiguiente, se ostenta que la mejor opción es la placa NODEMCU con un **52,6168 %** que se utilizará en la fase de pruebas e implementación. Es importante resaltar que esta técnica se aplicará en todas las tablas que contengan los criterios y opciones para concluir la opción idónea.

2.1.3. Análisis tarjeta de desarrollo ESP-32.

Se detallarán los prototipos existentes para este dispositivo inalámbrico. Además, se resalta que es la versión mejorada de la popular tarjeta de desarrollo ESP-8266, elemento viable para proyectos de domótica.

2.1.3.1. *ESP-WROVER-KIT-VB.*

Según Espressif Inc (2019), ESP-WROVER-KIT-VB incorpora Flash y PSRAM, al igual que, una memoria RAM de 4,5 Megabytes y un procesador con dos núcleos de 240 MHz; con esta placa es posible crear cámaras de Internet, televisores avanzados o radios con conexión a Internet vinculando pantallas LCD, micrófonos y códecs (Espressif Inc, 2019f), misma que será mostrada en la Figura 6.



2.1.3.2. *ESP32-LyraT.*

ESP32-LyraT (Figura 7) es una tarjeta de código abierto para el Ámbito de Desarrollo para Audio (ADF) de Espressif Systems diseñada para altavoces y aplicaciones de hogares inteligentes. En otras palabras, es un dispositivo que permite realizar una red sin cables físicos de sensores con tecnología BLE (Bluetooth Low Energy), y también configurar una red inteligente conocida como WeChat (Espressif Inc, 2019b, pp. 1-3).

ESP32-LyraT

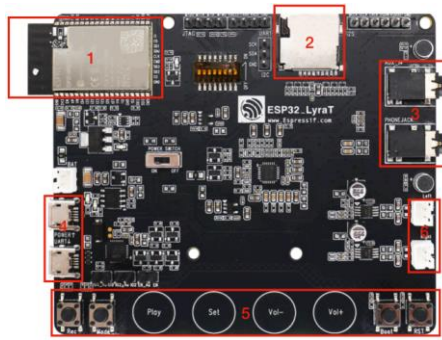


Figura 7. Placa ESP32-LyraT y sus componentes

Fuente: (Espressif Inc, 2019b, pp. 2-3)

Componentes de tarjeta de desarrollo ESP32.LyraT

- 1) Placa ESP32-WROVER-B
- 2) Tarjeta Micro SD
- 3) Salida de audio AUX_IN
- 4) Interfaz USB-UART e interfaz de alimentación USB
- 5) Botones táctiles
- 6) Interfaces de altavoz

2.1.3.3. ESP32-DevKitC.

La placa de la Figura 8, posee un diseño pequeño, botones de reinicio y flash, alimentación externa de 5 Voltios, puerto micro USB que funciona como el medio para poder comunicar el hardware con el software, además, casi todos sus pines de Entrada/Salida cuentan con encabezados separados de los pines para su fácil identificación. Asimismo, la placa ESP32-DevKitC es compatible con los dispositivos de las series ESP32-WROOM-32D, ESP32-WROOM-32U, ESP32-WROVER-B y ESP32-SOLO-1, gracias a su rendimiento óptimo con un conjunto completo de periféricos, Wi-Fi y conectividad Bluetooth (Espressif Inc, 2019a).

ESP32-DevKitC

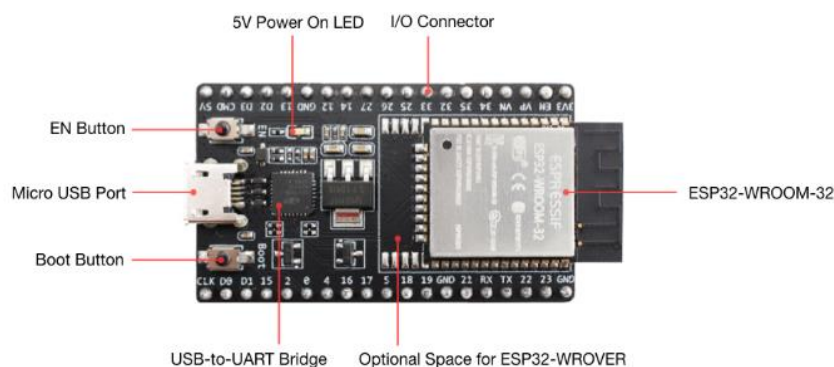


Figura 8. Placa ESP32-DevKitC y sus componentes
Fuente: (Espressif Inc, 2019a)

2.1.3.4. Análisis de priorización para placas ESP-32.

La Tabla 10, presentará las características de los elementos investigados y relacionados con las placas ESP32 y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 10. Características para opciones de placa ESP-32

Parámetro	ESP-WROVER	ESP32-LyraT	ESP32-DevKitC
CPU	Two Low-Power Xtensa® 32-bit LX6	Two Low-Power Xtensa® 32-bit LX6	Dual Core Xtensa® 32-bit LX6
Memoria Flash	4 MB	4 MB	4 MB
Velocidad Reloj	80 – 240 [MHz]	160 – 240 [MHz]	160 – 240 [MHz]
Alimentación	3,3 – 5 [V]	5 [V]	3,3 – 5 [V]
Conector micro USB	Tipo A, Tipo B	Tipo A, Tipo B	Tipo B
Protocolo WiFi	802,11 b/g/n/e/i	802,11 b/g/n	802,11 b/g/n/e/i
Protocolos de Seguridad Inalámbrica	WPA/WPA2/WPA2-Enterprise/WPS	WEP/WPA/WPA2	WPA/WPA2/WPA2-Enterprise/WPS
Conexión Bluetooth	Bluetooth v4,2 BR/EDR, Low Energy BLE	Clásico BT, Low Energy BLE	Clásico BT, Low Energy BLE
Compatibilidad Software	ESP-IDF	IoT Espressif App, ESP-ADF, ESP-IDF, Alexa	Arduino IDE, ESP-IDF, ESP-ADF, Eclipse
Precio	40 USD	28,53 USD	15 USD
Dimensiones	18*31*3,3 [mm]	95,5*80,6 [mm]	55*28*12,3 [mm]

Nota: Esta tabla contiene las características principales de las placas analizadas para realizar la priorización o selección para la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 11, contiene criterios de precio, capacidad de procesamiento, nivel de respuesta y disponibilidad; estos serán argumentados mediante el uso de la Tabla 10, misma que contiene detalles específicos de cada producto.

Tabla 11. Criterios y Opciones

CRITERIOS	OPCIONES
PRECIO	ESP-WROVER ESP32-LYRAT ESP32-DEVKITC
CAPACIDAD DE PROCESAMIENTO	
NIVEL DE RESPUESTA	
DISPONIBILIDAD	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas de la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 12. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Precio	Capacidad procesamiento	Nivel respuesta	Disponibilidad	Valor	Ponderado definido
1	Precio		10	1	5	16	0,604
2	Capacidad procesamiento	1/10		5	1	6,1	0,230
3	Nivel de respuesta	1	1/5		1	2,2	0,083
4	Disponibilidad	1/5	1	1		2,2	0,083
					TOTAL	26,5	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 12, realizará el mismo proceso desarrollado en la Tabla 4 (Criterios de Priorización); de la misma manera que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 13, 14, 15 y 16, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 13. Matriz Precio

#	A	B	C	D	E	F
0	PRECIO	ESP-WROVER	ESP32-LYRAT	ESP32-DEVKITC	VALOR	Ponderado relativo
1	ESP-WROVER		1/5	1/10	0,3	0,014
2	ESP32-LYRAT	5		1/5	5,2	0,254
3	ESP32-DEVKITC	10	5		15	0,732
				TOTAL	20,5	1

Nota: Esta tabla compara las opciones con relación al criterio Precio de la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 14. Matriz Capacidad de Procesamiento

#	A	B	C	D	E	F
0	CAPACIDAD DE PROCESAMIENTO	ESP-WROVER	ESP32-LYRAT	ESP32-DEVKITC	VALOR	Ponderado relativo
1	ESP-WROVER		1/5	1/5	0,4	0,032
2	ESP32-LYRAT	5		1	6	0,484
3	ESP32-DEVKITC	5	1		6	0,484
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Capacidad de procesamiento de la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 15. Matriz Nivel de Respuesta

#	A	B	C	D	E	F
0	NIVEL DE RESPUESTA	ESP-WROVER	ESP32-LYRAT	ESP32-DEVKITC	VALOR	Ponderado relativo
1	ESP-WROVER		1/5	1/5	0,4	0,032
2	ESP32-LYRAT	5		1	6	0,484
3	ESP32-DEVKITC	5	1		6	0,484
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Nivel de respuesta de la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 16. Matriz Disponibilidad

#	A	B	C	D	E	F
0	DISPONIBILIDAD	ESP-WROVER	ESP32-LYRAT	ESP32-DEVKITC	VALOR	Ponderado relativo
1	ESP-WROVER		1	1/10	1,1	0,050
2	ESP32-LYRAT	1		1/10	1,1	0,050
3	ESP32-DEVKITC	10	10		20	0,900
				TOTAL	22,2	1

Nota: Esta tabla compara las opciones con relación al criterio Disponibilidad de la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 17. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	PRECIO	CAPACIDAD DE PROCESAMIENTO	NIVEL DE RESPUESTA	DISPONIBILIDAD	PONDERADO
1	ESP-WROVER	0,604*0,014	0,230*0,032	0,083*0,032	0,083*0,050	0,02262
2	ESP32-LYRAT	0,604*0,254	0,230*0,484	0,083*0,484	0,083*0,050	0,30906
3	ESP32-DEVKITC	0,604*0,732	0,230*0,484	0,083*0,484	0,083*0,900	0,66832
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción para la placa ESP-32.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 17, desarrollará el mismo proceso que la Tabla 9 (Matriz Criterios vs Opciones), por consiguiente, se ostenta que la mejor opción es la placa ESP32-DevKitC con un **66,832 %** para ser utilizado en la fase de pruebas e implementación.

2.1.4. Análisis sensor de presencia.

Dispositivos pequeños que se utilizan para detectar movimientos físicos en temas vinculados con la seguridad.

2.1.4.1. Sensor PIR.

El sensor PIR HC-SR501 es accesible por su bajo costo y tamaño, siendo una opción que destaca gracias a su tecnología de punta referente a sensores de movimiento, además, cuenta

con 2 potenciómetros y 1 jumper que brindan la potestad de alterar sus parámetros de acuerdo a las necesidades de cada aplicación (Punto Flotante S.A., 2017, pág. 4).

2.1.4.2. Sensor de movimiento por Radar HB-100.

La Figura 9, representa el módulo transceptor Doppler biestático de banda X, además, es un sensor ideal para reducir falsas alarmas en detectores de intrusión junto con un sensor PIR. También es utilizado para aplicaciones que permiten la apertura de puertas y la medición de velocidad para un vehículo automóvil (ST Engineering, 2018).

Sensor Microondas HB100



Figura 9. Sensor Microondas HB100, vista frontal y posterior
Fuente: (ST Engineering, 2018)

2.1.4.3. Sensor ultrasónico HC-SR04 P.

Es un sensor de proximidad utilizado en proyectos de robótica como la detección del nivel de agua o generalmente como un sensor de estacionamiento; posee un sonar, el cual transmite un pulso ultrasónico ($\sim 40\text{KHz}$) desde la unidad hasta el objetivo determinando la distancia existente tras conocer el tiempo requerido para el retorno del eco (Handson Technology, 2018, pág. 4), mismo que será mostrado en la Figura 10.

Sensor ultrasónico HC-SR04 P



Figura 10. Sensor ultrasónico que detecta por medio de un sonar.
Fuente: (Handson Technology, 2018, pág. 1)

2.1.4.4. Análisis de priorización para sensor de presencia.

En esta sección se presentarán las características (mostradas en la Tabla 18) de los elementos investigados y relacionados con los sensores detectores de movimiento/presencia y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 18. Características para opciones de sensor de presencia

Parámetro	HC-SR501	HB-100	HC-SR04 P
Alimentación	5 – 12 [V]	5 [V]	3,3 – 5 [V]
Ajuste para tiempo de detección	3s hasta 5min	4 μ s hasta 10 μ s	10 μ s hasta 50 μ s
Rango de detección	3 - 7 [m]	8 [m]	0,02 - 4,5 [m]
Ángulo de detección	120°	360°	15°
Tipo de detección	Rayos infrarrojos	Señales microondas	Ultrasonido
Tiempo de inicialización	1 [min]	3 [min]	2 [min]
Precio	3,50 USD	25 USD	3,50 USD
Dimensiones	32*24*18 [mm]	38*45*7 [mm]	73*17 [mm]

Nota: Esta tabla contiene las características principales de los sensores analizados para realizar la priorización o selección del sensor de presencia.

Elaborado por: Juan Fonseca y Daniel Soria

Para realizar un análisis de priorización será necesario la Tabla 19, misma que contiene criterios de tiempo de detección, rango de detección, precio y tiempo de inicialización; estos serán argumentados mediante el uso de la Tabla 18, misma que contiene detalles específicos de cada producto.

Tabla 19. Criterios y Opciones

CRITERIOS	OPCIONES
TIEMPO DE DETECCIÓN	PIR HC-SR501 HB-100 HC-SR04 P
RANGO DE DETECCIÓN	
PRECIO	
TIEMPO DE INICIALIZACIÓN	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas del sensor de presencia.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 20, desarrollará el mismo proceso realizado en la Tabla 4 (Criterios de Priorización); de la misma forma que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 21, 22, 23 y 24, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 20. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Tiempo detección	Rango detección	Precio	Tiempo inicialización	Valor	Ponderado definido
1	Tiempo de detección		5	1/5	5	10,2	0,364
2	Rango de detección	1/5		1/5	1	1,4	0,050
3	Precio	5	5		5	15	0,536
4	Tiempo de inicialización	1/5	1	1/5		1,4	0,050
					TOTAL	28	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para el sensor de presencia.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 21. Matriz Tiempo de Detección

#	A	B	C	D	E	F
0	TIEMPO DE DETECCIÓN	HC-SR501	HB-100	HC-SR04 P	VALOR	Ponderado relativo
1	HC – SR501		10	5	15	0,732
2	HB – 100	1/10		1/5	0,3	0,014
3	HC – SR04 P	1/5	5		5,2	0,254
				TOTAL	20,5	1

Nota: Esta tabla compara las opciones con relación al criterio Tiempo de Detección del sensor de presencia.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 22. Matriz Rango de Detección

#	A	B	C	D	E	F
0	RANGO DE DETECCIÓN	HC-SR501	HB-100	HC-SR04 P	VALOR	Ponderado relativo
1	HC – SR501		1	5	6	0,484
2	HB – 100	1		5	6	0,484
3	HC – SR04 P	1/5	1/5		0,4	0,032
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Rango de Detección del sensor de presencia.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 23. Matriz Precio

#	A	B	C	D	E	F
0	PRECIO	HC-SR501	HB-100	HC-SR04 P	VALOR	Ponderado relativo
1	HC – SR501		10	1	11	0.495
2	HB – 100	1/10		1/10	0.2	0.010
3	HC – SR04 P	1	10		11	0.495
				TOTAL	22.2	1

Nota: Esta tabla compara las opciones con relación al criterio Precio del sensor de presencia.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 24. Matriz Tiempo de Inicialización

#	A	B	C	D	E	F
0	TIEMPO DE INICIALIZACIÓN	HC-SR501	HB-100	HC-SR04 P	VALOR	Ponderado relativo
1	HC – SR501		10	5	15	0,591
2	HB – 100	1/10		1/10	0,2	0,007
3	HC – SR04 P	1/5	10		10,2	0,402
				TOTAL	25,4	1

Nota: Esta tabla compara las opciones con relación al criterio Tiempo de Inicialización del sensor de presencia.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 25. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	TIEMPO DE DETECCIÓN	RANGO DE DETECCIÓN	PRECIO	TIEMPO DE INICIALIZACIÓN	PONDERADO
1	HC – SR501	0,364*0,732	0,050*0,484	0,536*0,495	0,050*0,591	0,585518
2	HB – 100	0,364*0,014	0,050*0,484	0,536*0,010	0,050*0,007	0,035006
3	HC – SR04 P	0,364*0,254	0,050*0,032	0,536*0,495	0,050*0,402	0,379476
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción para el sensor de presencia.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 25, desarrollará el mismo proceso que la Tabla 9 (Matriz Criterios vs Opciones), por consiguiente, se ostenta que la mejor opción es el sensor PIR HC-SR501 con un **58,55%** para ser utilizado en la fase de pruebas e implementación.

2.1.5. Análisis sensor de temperatura/humedad.

Dispositivos pequeños que se utilizan para la detección de temperatura/humedad mediante lecturas del ambiente.

2.1.5.1. Sensor DHT11.

El sensor DHT11 (Figura 11) mide la temperatura o la humedad gracias a su exclusiva técnica de adquisición para señales digitales, garantizando duración y confiabilidad a largo plazo (Mouser Electronics, 2019).

Sensor DHT11



Figura 11. Sensor para detectar temperatura y/o humedad DHT11
Fuente: (Mouser Electronics, 2019, pág. 3)

2.1.5.2. Sensor DHT22.

El sensor DHT22 mostrado en la Figura 12, cuenta con una señal digital calibrada gracias a la tecnología exclusiva para la señal digital y detección de humedad, certificando su confiabilidad y estabilidad. Además, “los sensores del módulo son compensados por temperatura y calibrados en una cámara de medición exacta y así, el coeficiente de calibración

se guarda en la memoria OTP”, mostrando el coeficiente de la memoria cuando el sensor se activa (Aosong Electronics Co., s. f.).

Sensor DHT22 o AM2302

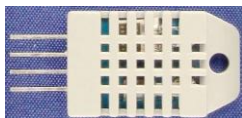


Figura 12. Sensor para detectar temperatura y/o humedad DHT22
Fuente: (Aosong Electronics Co., s. f.)

2.1.5.3. Análisis de priorización para sensor de temperatura/humedad.

La Tabla 26, presentará las características de los elementos investigados y relacionados con los sensores detectores de temperatura - humedad y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 26. Características para opciones de sensor de temperatura/humedad

Parámetro	DHT11	DHT22
Alimentación	3 – 5 [V]	3,3 – 6 [V]
Señal de salida	Digital	Digital
Rango de medición de temperatura	0 a 50 [°C]	-40 a 80 [°C]
Precisión de medición de temperatura	±2,0 [°C]	±0,5 [°C]
Resolución Temperatura	0,1 [°C]	0,1 [°C]
Rango de medición para humedad	20% - 90% [RH]	0% - 100% [RH]
Precisión de medición para humedad	4% [RH]	2% [RH]
Resolución Humedad	1% [RH]	0,1% [RH]
Tiempo de muestreo	2 [s]	2 [s]
Precio	3 USD	10 USD
Dimensiones	16*12*5 [mm]	20*15*8 [mm]

Nota: Esta tabla contiene las características principales de los sensores analizados para realizar la priorización o selección del sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

Para realizar un análisis de priorización será necesario la Tabla 27, misma que contiene criterios de precio, rango de medición, precisión de medición y tiempo de muestreo; estos serán argumentados mediante el uso de la Tabla 26, misma que contiene detalles específicos de cada producto.

Tabla 27. Criterios y Opciones

CRITERIOS	OPCIONES
PRECIO	DHT11 DHT22
RANGO DE DETECCIÓN	
PRECISIÓN DE MEDICIÓN	
TIEMPO DE MUESTREO	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas del sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 28, desarrollará el mismo proceso realizado en la Tabla 4 (Criterios de Priorización); de la misma forma que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 29, 30, 31 y 32, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 28. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Precio	Rango detección	Precisión medición	Tiempo muestreo	Valor	Ponderado definido
1	Precio		10	10	10	30	0,826
2	Rango detección	1/10		1	1	2,1	0,058
3	Precisión medición	1/10	1		1	2,1	0,058
4	Tiempo muestreo	1/10	1	1		2,1	0,058
					TOTAL	36,3	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para el sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 29. Matriz Precio

#	A	B	C	D	E
0	PRECIO	DHT11	DHT22	VALOR	Ponderado relativo
1	DHT11		10	10	0,990
2	DHT22	1/10		0,1	0,010
			TOTAL	10,1	1

Nota: Esta tabla compara las opciones con relación al criterio Precio del sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 30. Matriz Rango de Detección

#	A	B	C	D	E
0	RANGO DE DETECCIÓN	DHT11	DHT22	VALOR	Ponderado relativo
1	DHT11		1/5	0,2	0,038
2	DHT22	5		5	0,962
			TOTAL	5,2	1

Nota: Esta tabla compara las opciones con relación al criterio Rango de Detección del sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 31. Matriz Precisión de Medición

#	A	B	C	D	E
0	PRECISIÓN DE MEDICIÓN	DHT11	DHT22	VALOR	Ponderado relativo
1	DHT11		1/5	0,2	0,038
2	DHT22	5		5	0,962
			TOTAL	5,2	1

Nota: Esta tabla compara las opciones con relación al criterio Precisión de Medición del sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 32. Matriz Tiempo de Muestreo

#	A	B	C	E	F
0	TIEMPO DE MUESTREO	DHT11	DHT22	VALOR	Ponderado relativo
1	DHT11		1	1	0,500
2	DHT22	1		1	0,500
			TOTAL	2	1

Nota: Esta tabla compara las opciones con relación al criterio Tiempo de Muestreo del sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 33. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	PRECIO	RANGO DETECCIÓN	PRECISIÓN MEDICIÓN	TIEMPO MUESTREO	PONDERADO
1	DHT11	0,826*0,990	0,058*0,038	0,058*0,038	0,058*0,500	0,851148
2	DHT22	0,826*0,010	0,058*0,962	0,058*0,962	0,058*0,500	0,148852
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción para el sensor de temperatura/humedad.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 33, desarrollará el mismo proceso que la Tabla 9 (Matriz Criterios vs Opciones), por consiguiente, se ostenta que la mejor opción es el sensor de temperatura y humedad DHT-11 con un **85,11%** para ser utilizado en la fase de pruebas e implementación.

2.1.6. Análisis sensor de gas/humo.

Dispositivos pequeños que se utilizan para detectar gases inflamables como GLP, metano, alcohol, butano, entre otros. Este sensor es idóneo para implementarlo en proyectos de domótica que requieran seguridad para la integridad de los habitantes de una vivienda.

2.1.6.1. Sensor MQ-2.

El sensor MQ-2 tiene la función principal de detectar fugas de gas combustible y humo, posee alta sensibilidad ante GLP, propano e hidrógeno. Además, tiene un precio bajo y se adapta en diferentes aplicaciones de seguridad relacionadas con el ambiente. En otras palabras, sus

prestaciones permiten una respuesta rápida, estabilidad, un rango extenso en su detección, entre otros (Olimex, 2019). La Figura 13, muestra el sensor mencionado para la detección de gas.

Sensor de gas MQ-2



Figura 13. Sensor para detectar gas MQ-2
Fuente: (Olimex, 2019)

2.1.6.2. Sensor MQ-5.

El sensor MQ-5 (Figura 14) es útil principalmente para alertar fugas de gas. Una de las características destacadas es su potenciómetro incorporado, el cual permite ajustar su sensibilidad brindando la posibilidad de detectar GLP, gas butano, gas carbón, metano, humo, entre otros. Además, su voltaje de salida se incrementa a la par con los gases detectados (Waveshare, 2019).

Sensor de gas MQ-5



Figura 14. Sensor para detectar gas MQ-5
Fuente: (Waveshare, 2019)

2.1.6.3. Sensor MQ-6.

El elemento mostrado en la Figura 15, detecta diferentes tipos de gases inflamables, sin embargo, posee alta sensibilidad al gas propano GLP. Las aplicaciones principales en las que es ampliamente utilizado este sensor son alarmas de fuga de gas doméstico, alarmas de gas inflamable industrial y detectores de gas portátil (Zhengzhou Winsen Electronics Technology CO., LTD, s. f.).

Sensor de gas MQ-6



Figura 15. Sensor para gases inflamables MQ-6

Fuente: (Zhengzhou Winsen Electronics Technology CO., LTD, s. f.)

2.1.6.4. Análisis de priorización para sensor de gas/humo.

La Tabla 34, presentará las características de los elementos investigados y relacionados con los sensores detectores de gas/humo y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 34. Características para opciones de sensor de gas/humo

Parámetro	MQ-2	MQ-5	MQ-6
Alimentación	5 [V]	5 [V]	5 [V]
Señal de salida	Digital	Analógica / Digital	Analógica / Digital
Rango de detección	300-10000 [ppm]	200-10000 [ppm]	200-10000 [ppm]
Gas característico	Gas combustible, isobutano (humo)	Gas licuado, gas natural, gas propano	Gas propano y butano (LPG), gas natural (metano), gas i-butano
Tiempo de precalentamiento	48 [h]	24 [h]	24 [h]
Tiempo de Respuesta	≤ 10 [s]	≤ 10 [s]	≤ 10 [s]
Tiempo de recuperación	≤ 30 [s]	≤ 30 [s]	≤ 30 [s]
Precio	5 USD	5 USD	4,5 USD
Dimensiones	36,4*26,6 [mm]	40*21 [mm]	40*20 [mm]

Nota: Esta tabla contiene las características principales de los sensores analizados para realizar la priorización o selección del sensor de gas/humo.

Elaborado por: Juan Fonseca y Daniel Soria

Para realizar un análisis de priorización será necesario la Tabla 35, misma que contiene criterios de precio, rango de detección, gas característico y tiempo de precalentamiento; estos serán argumentados mediante el uso de la Tabla 34, misma que contiene detalles específicos de cada producto.

Tabla 35. Criterios y Opciones

CRITERIOS	OPCIONES
PRECIO	MQ-2 MQ-5 MQ-6
RANGO DE DETECCIÓN	
GAS CARACTERÍSTICO	
TIEMPO DE PRECALENTAMIENTO	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas del sensor de gas/humo.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 36, desarrollará el mismo proceso realizado en la Tabla 4 (Criterios de Priorización); de la misma forma que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 37, 38, 39 y 40, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 36. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Precio	Rango detección	Gas característico	Tiempo precalentamiento	Valor	Ponderado definido
1	Precio		5	1	5	11	0,370
2	Rango detección	1/5		1/10	1/5	0,5	0,017
3	Gas característico	1	10		1	12	0,404
4	Tiempo precalentamiento.	1/5	5	1		6,2	0,209
					TOTAL	29,7	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para el sensor de gas/humo.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 37. Matriz Precio

#	A	B	C	D	E	F
0	PRECIO	MQ-2	MQ-5	MQ-6	VALOR	Ponderado relativo
1	MQ-2		1	1/5	1,2	0,097
2	MQ-5	1		1/5	1,2	0,097
3	MQ-6	5	5		10	0,806
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Precio del sensor de gas/humo.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 38. Matriz Rango de Detección

#	A	B	C	D	E	F
0	RANGO DE DETECCIÓN	MQ-2	MQ-5	MQ-6	VALOR	Ponderado relativo
1	MQ-2		1/5	1/5	0,4	0,032
2	MQ-5	5		1	6	0,484
3	MQ-6	5	1		6	0,484
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Rango de Detección del sensor de gas/humo.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 39. Matriz Gas Característico

#	A	B	C	D	E	F
0	GAS CARACTERISTICO	MQ-2	MQ-5	MQ-6	VALOR	Ponderado relativo
1	MQ-2		1/5	1/10	0,3	0,014
2	MQ-5	5		1/5	5,2	0,254
3	MQ-6	10	5		15	0,732
				TOTAL	20,5	1

Nota: Esta tabla compara las opciones con relación al criterio Gas Característico del sensor de gas/humo.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 40. Matriz Tiempo de Pre calentamiento

#	A	B	C	D	E	F
0	TIEMPO DE PRECALENTAMIENTO	MQ-2	MQ-5	MQ-6	VALOR	Ponderado relativo
1	MQ-2		1/5	1/5	0,4	0,032
2	MQ-5	5		1	6	0,484
3	MQ-6	5	1		6	0,484
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Tiempo de Pre calentamiento del sensor de gas/humo.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 41. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	PRECIO	RANGO DETECCIÓN	GAS CARÁCT,	TIEMPO PRECALENTAM,	PONDERADO
1	MQ-2	0,370*0,097	0,017*0,032	0,404*0,014	0,209*0,032	0,048778
2	MQ-5	0,370*0,097	0,017*0,484	0,404*0,254	0,209*0,484	0,24789
3	MQ-6	0,370*0,806	0,017*0,484	0,404*0,732	0,209*0,484	0,703332
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción para el sensor de gas/humo.
Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 41, desarrollará el mismo proceso que la Tabla 9 (Matriz Criterios vs Opciones), por consiguiente, se ostenta que la mejor opción es el sensor MQ-6 con un **70,33 %** para ser utilizado en la fase de pruebas e implementación.

2.1.7. Análisis motor para movimiento de puerta y persiana.

Dispositivos con un tamaño pequeño utilizados para el movimiento de elementos en proyectos de robótica y domótica. A pesar de su potencia y velocidad limitada, es apto para prototipos experimentales o maquetas demostrativas.

2.1.7.1. Motor modelo 28BYJ-48 paso a paso.

El dispositivo mostrado en la Figura 16, es utilizado generalmente en proyectos de domótica gracias a su mínimo costo económico, cuenta con cuatro estados que actúan de acuerdo a la excitación de cada bobina generando “un paso”. Además, cuenta con una alimentación de 5V haciendo de este elemento una opción idónea para realizar simulaciones a menor escala (Prometec, s. f.).

Motor 28BYJ-48 y chip ULN2003A



Figura 16. Motor de 5 voltios junto a su chip para secuencia de pasos
Fuente: (Prometec, s. f.)

2.1.7.2. Motor DC básico 5 voltios.

El motor DC (Figura 17) conocido también como motor de juguete es utilizado generalmente en proyectos de electrónica debido a su resistencia, facilidad de uso y control y cuenta con una gama amplia en nivel de voltaje. Este motor puede girar e invertir su dirección de movimiento dependiendo del circuito elaborado. Además, posee compatibilidad con Arduino y Adafruit Motor Shield permitiendo disponibilidad inmediata para desarrollar prototipos didácticos (COMPONENTS101, 2018).

Motor DC de juguete

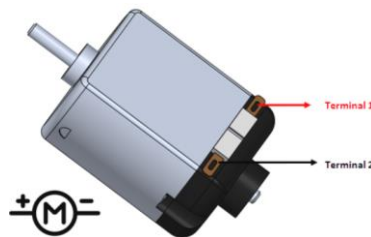


Figura 17. Motor DC y cableado
Fuente: (COMPONENTS101, 2018)

2.1.7.3. Servomotor SG-90.

El dispositivo de la Figura 18, es pequeño y ligero que gira alrededor de 180° (90° horario o antihorario), trabajando de manera semejante a un modelo estándar, pero en una versión reducida; también se acopla con cualquier código de servo, hardware o librería para controlar este elemento (COMPONENTS101, 2017).

Servomotor SG-90



Figura 18. Servomotor SG 90
Fuente: (COMPONENTS101, 2017)

2.1.7.4. Análisis de priorización para motor de puerta y persiana.

La Tabla 42, presentará las características de los elementos investigados y relacionados con los motores para movimiento de puertas/persianas y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 42. Características para opciones de motor de puerta y persiana

Parámetro	Motor 28BYJ-48	Motor DC	Motor SG-90
Alimentación	5 - 12 [V]	5 [V]	5 [V]
Señal de salida	Digital	Analógica / Digital	Analógica / Digital
Consumo de corriente	200 [mA]	600 [mA]	350 [mA]
Rotación	360° (ambos sentidos)	360° (ambos sentidos)	90° (cada sentido)
Potencia máxima	20 [W]	50 [W]	2,13 [W]
Peso	35 [g]	17,5 [g]	9 [g]
Precio	6,5 USD	6,5 USD	4,5 USD
Dimensiones	31*27*19 [mm]	27,5*20*15 [mm]	22,2*11,8*31 [mm]

Nota: Esta tabla contiene las características principales de los motores analizados para realizar la priorización o selección.

Elaborado por: Juan Fonseca y Daniel Soria

Para realizar un análisis de priorización será necesario la Tabla 43, misma que contiene criterios de precio, rotación, alimentación y precisión y control de movimiento; estos serán argumentados mediante el uso de la Tabla 42, misma que contiene detalles específicos de cada producto.

Tabla 43. Criterios y Opciones

CRITERIOS	OPCIONES
PRECIO	28BYJ-48 DC SG-90
ROTACIÓN	
ALIMENTACIÓN	
PRECISIÓN Y CONTROL DE MOVIMIENTO	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas del motor de puerta y persiana.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 44, desarrollará el mismo proceso realizado en la Tabla 4 (Criterios de Priorización); de la misma forma que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 45, 46, 47 y 48, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 44. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Precio	Rotación	Alimentación	Precisión & control movimiento	Valor	Ponderado definido
1	Precio		1/5	1/5	1/10	0,5	0,015
2	Rotación	5		1	1/5	6,2	0,189
3	Alimentación	5	1		1/5	6,2	0,189
4	Precisión & control movimiento	10	5	5		20	0,607
					TOTAL	32,9	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para el motor de puerta y persiana.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 45. Matriz Precio

#	A	B	C	D	E	F
0	PRECIO	28BYJ-48	DC	SG-90	VALOR	Ponderado relativo
1	28BYJ-48		1	1/5	1,2	0,097
2	DC	1		1/5	1,2	0,097
3	SG-90	5	5		10	0,806
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Precio del motor de puerta y persiana.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 46. Matriz Rotación

#	A	B	C	D	E	F
0	ROTACIÓN	28BYJ-48	DC	SG-90	VALOR	Ponderado relativo
1	28BYJ-48		1	5	6	0,484
2	DC	1		5	6	0,484
3	SG-90	1/5	1/5		0,4	0,032
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Rotación del motor de puerta y persiana.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 47. Matriz Alimentación

#	A	B	C	D	E	F
0	ALIMENTACIÓN	28BYJ-48	DC	SG-90	VALOR	Ponderado relativo
1	28BYJ-48		5	5	10	0,806
2	DC	1/5		1	1,2	0,097
3	SG-90	1/5	1		1,2	0,097
				TOTAL	12,4	1

Nota: Esta tabla compara las opciones con relación al criterio Alimentación del motor de puerta y persiana.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 48. Matriz Precisión & Control de Movimiento

#	A	B	C	D	E	F
0	PRECISION & CONTROL DE MOVIMIENTO	28BYJ-48	DC	SG-90	VALOR	Ponderado relativo
1	28BYJ-48		10	10	20	0,900
2	DC	1/10		1	1,1	0,050
3	SG-90	1/10	1		1,1	0,050
				TOTAL	22,2	1

Nota: Esta tabla compara las opciones con relación al criterio Precisión & Control de Movimiento del motor de puerta y persiana.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 49. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	PRECIO	ROTACIÓN	ALIMENTACIÓN	PRECISIÓN & CONTROL MOVIMIENTO	PONDERADO
1	28BYJ-48	0,015*0,097	0,189*0,484	0,189*0,806	0,607*0,900	0,791565
2	DC	0,015*0,097	0,189*0,484	0,189*0,097	0,607*0,050	0,141614
3	SG-90	0,015*0,806	0,189*0,032	0,189*0,097	0,607*0,050	0,066821
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción para el motor de puerta y persiana.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 49, desarrollará el mismo proceso que la Tabla 9 (Matriz Criterios vs Opciones), por consiguiente, se ostenta que la mejor opción es el motor modelo 28BYJ-48 paso a paso con un **79,16 %** para ser utilizado en la fase de pruebas e implementación.

2.1.8. Análisis cámara de video.

Son tarjetas de desarrollo con cámara de video integrada similares a la placa ESP-32 utilizadas para capturar imagen o video principalmente en proyectos relacionados con la seguridad del hogar. Además, es posible incrementar su almacenamiento interno gracias a su compatibilidad con tarjetas micro SD.

2.1.8.1. ESP32-CAM.

La Figura 19, es una placa que cuenta con una pequeña cámara que funciona independientemente gracias a su microcontrolador integrado ESP32, sus dimensiones son 27 mm * 40,5 mm * 4,5 mm y su corriente de 6mA. Además, este dispositivo se utiliza para monitoreo inalámbrico, identificación inalámbrica QR, señales de sistemas de posicionamiento inalámbrico y otras aplicaciones de domótica, siendo una solución ideal en caso de diseñar e implementar varios terminales de hardware de IoT (AI-Thinker, 2017).

ESP32-CAM



Figura 19. Placa ESP32-CAM, vista frontal y trasera
Fuente: (AI-Thinker, 2017)

2.1.8.2. ESP-EYE.

El dispositivo de la Figura 20, está diseñado para reconocer imágenes y procesar audio brindando la capacidad de ejecutarlo en varias aplicaciones IoT. Cuenta con un chip ESP32, una cámara de 2 megapíxeles y un micrófono. ESP-EYE ofrece almacenamiento externo, RAM de 8 Megabytes y flash de 4 Megabytes; además, este dispositivo transmite imágenes a través de Wi-Fi y utiliza depuración por medio de un puerto Micro-USB (Espressif Inc, 2019e).

ESP-EYE



Figura 20. ESP-EYE, vista frontal y trasera
Fuente: (Espressif Inc, 2019e)

2.1.8.3. Análisis de priorización para cámara de video.

La Tabla 50, presentará las características de los elementos investigados y relacionados con las placas que capturan imagen o video por cuestiones de seguridad y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 50. Características para opciones de cámara de video

Parámetro	ESP32-CAM	ESP-EYE
Alimentación	3 - 5 [V]	5 [V]
Señal de salida	Digital	Digital
Protocolos Inalámbricos	802,11 b/g/n 802,15 BR/EDR Low Energy BLE	802,11 b/g/n 802,15 BR/EDR
Cámaras compatibles	OV2640 / OV2670	OV2640
Conector micro USB	Tipo B	Tipo B
Socket TF Card	Si	No
Aplicaciones	Captura de imágenes y video en tiempo real	Reconocimiento facial
Peso	20 [g]	35,6 [g]
Precio	19,99 USD	31,99 USD
Dimensiones	27*40,5*4,5 [mm]	27,5*20*15 [mm]

Nota: Esta tabla contiene las características principales de las cámaras analizadas para realizar la priorización o selección.

Elaborado por: Juan Fonseca y Daniel Soria

Para realizar un análisis de priorización será necesario la Tabla 51, misma que contiene criterios de precio, compatibilidad de memoria externa, compatibilidad de cámaras y transmisión en tiempo real; estos serán argumentados mediante el uso de la Tabla 50, misma que contiene detalles específicos de cada producto.

Tabla 51. Criterios y Opciones

CRITERIOS	OPCIONES
PRECIO	ESP32-CAM ESP-EYE
COMPATIBILIDAD DE MEMORIA EXTERNA	
COMPATIBILIDAD DE CÁMARAS	
TRANSMISIÓN EN TIEMPO REAL	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas en la cámara de video.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 52, desarrollará el mismo proceso realizado en la Tabla 4 (Criterios de Priorización); de la misma forma que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 53, 54, 55 y 56, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 52. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Precio	Compatib, Memoria externa	Compatib, Cámaras	Transmisión Tiempo real	Valor	Ponderado definido
1	Precio		5	5	5	15	0,605
2	Compatib, Memoria externa	1/5		1	1	2,2	0,089
3	Compatib, Cámaras	1/5	1		1/5	1,4	0,056
4	Transmisión Tiempo real	1/5	1	5		6,2	0,250
					TOTAL	24,8	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para la cámara de video.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 53. Matriz Precio

#	A	B	C	D	E
0	PRECIO	ESP32-CAM	ESP-EYE	VALOR	Ponderado relativo
1	ESP32-CAM		5	5	0,962
2	ESP-EYE	1/5		0,2	0,038
			TOTAL	5,2	1

Nota: Esta tabla compara las opciones con relación al criterio Precio de la cámara de video.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 54. Matriz Compatibilidad De Memoria Externa

#	A	B	C	D	E
0	COMPATIBILIDAD DE MEMORIA EXTERNA	ESP32-CAM	ESP-EYE	VALOR	Ponderado relativo
1	ESP32-CAM		10	10	0,990
2	ESP-EYE	1/10		0,1	0,010
			TOTAL	10,1	1

Nota: Esta tabla compara las opciones con relación al criterio Compatibilidad de Memoria Externa de cámara de video.

Elaborado por: Juan Fonseca y Daniel Soria

Tabla 55. Matriz Compatibilidad De Cámaras

#	A	B	C	D	E
0	COMPATIBILIDAD DE CÁMARAS	ESP32-CAM	ESP-EYE	VALOR	Ponderado relativo
1	ESP32-CAM		5	5	0,962
2	ESP-EYE	1/5		0,2	0,038
			TOTAL	5,2	1

Nota: Esta tabla compara las opciones con relación al criterio Compatibilidad de Cámaras de la cámara de video.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 56. Matriz Transmisión En Tiempo Real

#	A	B	C	D	E
0	TRANSMISIÓN EN TIEMPO REAL	ESP32-CAM	ESP-EYE	VALOR	Ponderado relativo
1	ESP32-CAM		5	5	0,962
2	ESP-EYE	1/5		0,2	0,038
			TOTAL	5,2	1

Nota: Esta tabla compara las opciones con relación al criterio Transmisión en Tiempo Real para cámara de video.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 57. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	PRECIO	COMPATIB, MEMORIA EXTERNA	COMPATIB, CÁMARAS	TRANSM, TIEMPO REAL	PONDERADO
1	ESP32-CAM	0,605*0,962	0,089*0,990	0,056*0,962	0,250*0,962	0,964492
2	ESP-EYE	0,605*0,038	0,089*0,010	0,056*0,038	0,250*0,038	0,035508
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción para la cámara de video.
Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 57, desarrollará el mismo proceso que la Tabla 9 (Matriz Criterios vs Opciones), por consiguiente, se ostenta que la mejor opción es la placa ESP32-CAM con un **96,45 %** para ser utilizado en la fase de pruebas e implementación.

2.2. Análisis de Software

Para esta sección se realizan criterios de precio, compatibilidad de software con las tarjetas de desarrollo ESP8266 y ESP32, necesidad de plugin para obtener un funcionamiento ideal, tiempos de compilación y respuesta, entre otros.

2.2.1. Análisis software para programación de código.

2.2.1.1. *Arduino IDE.*

Arduino IDE es un programa informático utilizado para editar, compilar y depurar código mediante sus diferentes herramientas y funciones presentadas en un menú. Además, cuenta con un administrador de librerías y placas que permiten utilizar una diversidad amplia de dispositivos para su programación y demostración (Arduino, 2015).

2.2.1.2. *Arduino Create.*

Según Arduino (2020), Arduino Create es una plataforma en línea que escribe código, configura y comparte trabajos de manera similar a Arduino IDE. Además, es posible utilizar las tarjetas más populares de Arduino haciendo de este software un elemento viable en elaboración de proyectos. Sin embargo, varias de sus funciones suponen un valor económico (Arduino, 2020).

2.2.1.3. *ESPlorer IDE.*

Está diseñado para la tarjeta ESP8266. Es multiplataforma, es decir, es compatible con Windows, Linux, Solaris y Mac OS X. Además, este software de código abierto cuenta con una sintaxis que resalta LUA y Python permitiendo su edición y compilación (esp8266, 2014).

2.2.1.4. Análisis de priorización para software utilizado en desarrollo de código.

La Tabla 58, presentará las características de los elementos investigados y relacionados con los programas informáticos utilizados para editar y ejecutar código en las tarjetas de desarrollo mencionadas con anterioridad y así, concluir la opción idónea para el diseño, fase de pruebas e implementación.

Tabla 58. Características para opciones de programación del código

Parámetro	Arduino IDE	Arduino Create	ESPlorer IDE
Requiere plugin adicional	No	Si	Si
Compatibilidad con placa ESP8266	Si	Si	Si
Compatibilidad con placa ESP32	Si	No	No
Almacenamiento	Ilimitado	100 MB	Ilimitado
Tiempo de Compilación	Ilimitado	200 s / día	Ilimitado
Biblioteca Personalizada	Si	No	No
Fluidez de datos	Si	Si	No
Precio	Gratuito	6.99 USD / mes	Gratuito

Nota: Esta tabla contiene las características principales de los programas analizados para realizar la priorización o selección.

Elaborado por: Juan Fonseca y Daniel Soria

Para realizar un análisis de priorización será necesario la Tabla 59, misma que contiene criterios de compatibilidad con placas, tiempo de compilación, requisitos adicionales (plugin) y precio; estos serán argumentados mediante el uso de la Tabla 58, misma que contiene detalles específicos de cada producto.

Tabla 59. Criterios y Opciones

CRITERIOS	OPCIONES
COMPATIBILIDAD CON PLACAS	Arduino IDE Arduino Create ESPlorer IDE
TIEMPO DE COMPILACIÓN	
REQUISITOS ADICIONALES (PLUGIN)	
PRECIO	

Nota: Esta tabla contiene los criterios que resolverán la problemática del proyecto técnico, según su importancia, aplicando el proceso de priorización para las opciones propuestas del diseño del código.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 60, desarrollará el mismo proceso realizado en la Tabla 4 (Criterios de Priorización); de la misma forma que la Tabla 5, la pregunta de importancia se realizará en cada una de las Tablas 61, 62, 63 y 64, donde se tomará como referencia cada criterio utilizado según sea la tabla de priorización, realizando el mismo proceso que la Tabla 5.

Tabla 60. Criterios de Priorización

#	A	B	C	D	E	F	G
0	CRITERIO	Compatibilidad placas	Tiempo compilación	Plugin adicional	Precio	Valor	Ponderado definido
1	Compatibilidad		10	5	1	16	0,539
2	Tiempo compilación	1/10		5	5	10,1	0,340
3	Plugin adicional	1/5	1/5		1	1,4	0,047
4	Precio	1	1/5	1		2,2	0,074
					TOTAL	29,7	1

Nota: Esta tabla contiene los criterios de priorización acordados por el grupo de trabajo para el diseño del código.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 61. Matriz Compatibilidad

#	A	B	C	D	E	F
0	COMPATIBILIDAD	ARDUINO IDE	ARDUINO CREATE	ESPLORER IDE	VALOR	Ponderado relativo
1	ARDUINO IDE		10	10	20	0,900
2	ARDUINO CREATE	1/10		1	1,1	0,050
3	ESPLORER IDE	1/10	1		1,1	0,050
				TOTAL	22,2	1

Nota: Esta tabla compara las opciones con relación al criterio Compatibilidad del diseño del código.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 62. Matriz Tiempo de Compilación

#	A	B	C	D	E	F
0	TIEMPO DE COMPILACIÓN	ARDUINO IDE	ARDUINO CREATE	ESPLORER IDE	VALOR	Ponderado relativo
1	ARDUINO IDE		10	1	11	0,495
2	ARDUINO CREATE	1/10		1/10	0,2	0,010
3	ESPLORER IDE	1	10		11	0,495
				TOTAL	22,2	1

Nota: Esta tabla compara las opciones con relación al criterio Tiempo de Compilación del diseño del código.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 63. Matriz Plugin Adicional

#	A	B	C	D	E	F
0	PLUGIN ADICIONAL	ARDUINO IDE	ARDUINO CREATE	ESPLORER IDE	VALOR	Ponderado relativo
1	ARDUINO IDE		10	10	20	0,900
2	ARDUINO CREATE	1/10		1	1,1	0,050
3	ESPLORER IDE	1/10	1		1,1	0,050
				TOTAL	22,2	1

Nota: Esta tabla compara las opciones con relación al criterio Plugin Adicional del diseño del código.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 64. Matriz Precio

#	A	B	C	D	E	F
0	PRECIO	ARDUINO IDE	ARDUINO CREATE	ESPLORER IDE	VALOR	Ponderado relativo
1	ARDUINO IDE		10	1	11	0,495
2	ARDUINO CREATE	1/10		1/10	0,2	0,010
3	ESPLORER IDE	1	10		11	0,495
				TOTAL	22,2	1

Nota: Esta tabla compara las opciones con relación al criterio Precio del diseño del código.
Elaborado por: Juan Fonseca y Daniel Soria

Tabla 65. Matriz Criterios vs Opciones

#	A	B	C	D	E	F
0	CRITERIOS / OPCIONES	COMPATIBILIDAD PLACAS	TIEMPO COMPILACIÓN	PLUGIN ADICIONAL	PRECIO	PONDERADO
1	ARDUINO IDE	0,539*0,900	0,340*0,495	0,047*0,900	0,074*0,495	0,73233
2	ARDUINO CREATE	0,539*0,050	0,340*0,010	0,047*0,050	0,074*0,010	0,03344
3	ESPLORER IDE	0,539*0,050	0,340*0,495	0,047*0,050	0,074*0,495	0,23423
						1

Nota: Esta tabla contiene los cálculos que permiten conocer la mejor opción para el diseño del código.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 65, desarrollará el mismo proceso que la Tabla 9 (Matriz Criterios vs Opciones), por consiguiente, se ostenta que la mejor opción es el software de programación Arduino IDE con un **73,23 %** para ser utilizado en la fase de pruebas e implementación.

CAPÍTULO 3

3. Diseño y Desarrollo

En este apartado se diseñó y desarrolló los diagramas y códigos pertenecientes a cada elemento físico y/o lógico, para de esta manera proceder a la implementación de los mismos en un prototipo experimental. A continuación, se presentan los diagramas de los circuitos diseñados por el grupo de trabajo que se emplearon en la fase de pruebas.

3.1. Configuración General para Programación de Arduino

Para obtener un funcionamiento correcto de cualquier programa de Arduino es indispensable contar con un fichero principal que posea las funciones `setup()` y `loop()`. Estas funciones fueron aplicadas en cada “sketch” con sus respectivos cambios de variables e inicializaciones, siendo así repetitivas durante el desarrollo del código de este proyecto técnico.

Código para declaración de variables en Arduino IDE

```
//-----VARIABLES GLOBALES-----  
int contconexion = 0;  
  
const char *ssid = "NOMBRE_RED_WIFI";  
const char *password = "CONTRASEÑA_RED_WIFI";  
char SERVER[50] = "ml6.cloudmqtt.com";  
int SERVERPORT = 10770;  
String USERNAME = "controlador";  
char PASSWORD[50] = "admin.1";  
  
char PLACA[50];  
char valueStr[15];  
String strmq6 = "";  
char MQ6[50];  
  
char FOC03[50];  
char FOC04[50];  
char FOC05[50];
```

Figura 21. Declaración e inicialización de variables generales.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 21, muestra la declaración e inicialización de variables generales para la conexión WiFi, servidor, puerto, usuario y contraseña que comunicará la placa con la aplicación web.

Código para respuesta de servidor en Arduino IDE

```
void callback(char* topic, byte* payload, unsigned int length) {

    char PAYLOAD[5] = "    ";
    String mensaje;
    Serial.print("Mensaje Recibido: [");
    Serial.print(topic);
    Serial.print("] ");

    for (int i = 0; i < length; i++) {
        PAYLOAD[i] = (char)payload[i];
        mensaje += String(PAYLOAD[i]);
    }
    Serial.print("MENSAJE DEL SERVIDOR: ");
    Serial.println(mensaje);

    if (String(topic) == String(FOCO3)) {
        if (payload[1] == 'N') {
            digitalWrite(5, HIGH);
            Serial.print(payload[1]);
        }
    }
}
```

Figura 22. Comunicación con el servidor para interactuar con los dispositivos utilizados.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 22, muestra el método callback () con la comunicación y respuesta del servidor a un tópico, por medio de mensajes en tiempo real entre el servidor y los clientes.

Código para reconexión cliente - servidor en Arduino IDE

```
void reconnect() {
    uint8_t retries = 3;
    // Loop hasta que estamos conectados
    while (!client.connected()) {
        Serial.print("Intentando conexion MQTT...");
        // Crea un ID de cliente al azar
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        // Attempt to connect
        USERNAME.toCharArray(PLACA, 50);
        if (client.connect("", PLACA, PASSWORD)) {
            Serial.println("conectado");
            client.subscribe(FOCO3);
        }
    }
}
```

Figura 23. Código para reconectar en caso de perder la comunicación entre el servidor y cliente.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 23, muestra el código programado en las placas con el objetivo de reintentar cada cinco segundos una conexión con el servidor en caso de perderla hasta que sea exitosa.

Código para configuración de elementos en Arduino IDE

```
void setup() {  
  pinMode(IN1, OUTPUT);           // todos los pines c  
  
  pinMode(5, OUTPUT); // D1 salida focol  
  
  // Inicia Serial  
  Serial.begin(115200);  
  
  // Conexión WIFI  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED and contconexion < 50)  
    ++contconexion;  
    delay(500);  
    Serial.print(".");  
}
```

Figura 24. Configuración de sensores y actuadores de acuerdo al pin utilizado.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 24, muestra la asignación de pines analógicos o digitales (entrada/salida) para los diferentes sensores y actuadores utilizados en la programación de las placas, además, se reconectará automáticamente al perder conexión; mientras que, la Figura 25, representa la asignación de variables que se utilizó en el broker MQTT para una comunicación exitosa.

Código para configuración de variables en Arduino IDE con el servidor

```
client.setServer(SERVER, SERVERPORT);  
client.setCallback(callback);  
  
String sensor_MQ6 = "/" + USERNAME + "/" + "sensormq6";  
sensor_MQ6.toCharArray(MQ6, 50);  
String foco3 = "/" + USERNAME + "/" + "foco3";  
foco3.toCharArray(FOCO3, 50);  
String foco4 = "/" + USERNAME + "/" + "foco4";  
foco4.toCharArray(FOCO4, 50);  
String foco5 = "/" + USERNAME + "/" + "foco5";  
foco5.toCharArray(FOCO5, 50);
```

Figura 25. Configuración de variables para emparejar el cliente con el servidor
Elaborado por: Juan Fonseca y Daniel Soria

3.1.1. Librerías para programación del código.

Para programar conjuntamente las tarjetas de desarrollo, sensores y actuadores se utilizaron librerías necesarias para su correcto funcionamiento. Estas son descritas a continuación:

Librerías Wi-Fi

```
#include <ESP8266WiFi.h> → a
#include <WiFi.h> → b
```

Figura 26. Librerías para conexión inalámbrica Wi-Fi para tarjetas de desarrollos ESP
Elaborado por: Juan Fonseca y Daniel Soria

Se describen dos literales mostrados en la Figura 26:

a. Librería para establecer comunicación con una red Wi-Fi en tarjeta ESP-8266.

b. Librería para establecer comunicación con una red Wi-Fi en tarjeta ESP-32.

Por otra parte, la Figura 27, muestra la librería para suscribirse a uno o más tópicos como cliente MQTT-SN; a su vez, la Figura 28, muestra la librería utilizada para el sensor DHT en las placas ESP.

Librería cliente MQTT-SN

```
#include <PubSubClient.h>
```

Figura 27. Librería para suscribirse a uno o varios topics como cliente MQTT-SN.
Elaborado por: Juan Fonseca y Daniel Soria

Librería sensor de temperatura DHT

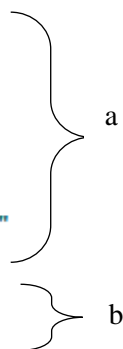
```
#include "DHTesp.h"
```

Figura 28. Librería de sensores de temperatura DHT para tarjetas ESP8266 y ESP32.
Elaborado por: Juan Fonseca y Daniel Soria

Librerías cámara de video ESP32-CAM

```
#include "esp_camera.h"
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_http_server.h"

#include "Base64.h"
```



a

b

Figura 29. Librerías para cargar y ejecutar la placa ESP32-CAM.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 29, describe dos literales:

- a. Librerías necesarias para establecer imagen y video en la tarjeta ESP32-CAM.
- b. Librería para capturar imagen y enviarla al servidor Google Drive en la tarjeta ESP32-CAM.

3.2. Diagramas para Módulos dentro del Hogar

A continuación, se presentan diagramas para la comunicación de las tarjetas de desarrollo ESP-8266, ESP-32 y ESP32-CAM con sus respectivos sensores y actuadores que se dispusieron en el prototipo experimental para digitalizar el control en su mayoría del hogar.

3.2.1. Dormitorios.

Para la sección dormitorios se utilizó una tarjeta de desarrollo ESP-8266, misma que se ubicó de manera estratégica para poder comunicar el cuarto uno y cuarto dos. Además, cada sector del módulo posee luces independientes, su respectivo sensor de luz que confirma si existe o no un foco encendido y habilita la función de abrir y cerrar una persiana ubicada en el dormitorio uno por medio del motor modelo 28BYJ-48 paso a paso. La Figura 30, muestra el diagrama del circuito utilizado en este módulo, mismo que consta de: una placa NodeMCU, un motor paso a paso, una placa integrada ULN2003, 3 leds, resistencia de 330 K Ω , un LDR y un sistema compuesto de un foco de 110 [V], un relé que transformó este voltaje y se adaptó a la alimentación de 5 [V] de la placa NodeMCU v3.

Circuito para módulo Dormitorios

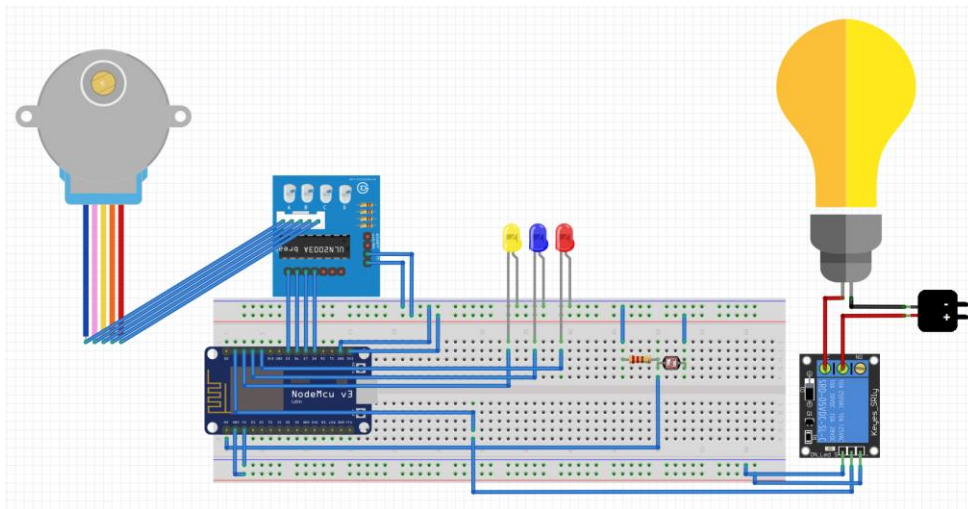


Figura 30. Diagrama del circuito para el módulo dormitorio uno y dos con sus respectivos actuadores
Elaborado por: Juan Fonseca y Daniel Soria

3.2.1.1. Código Dormitorios.

Se describe el código relevante que se utilizó para configurar y ejecutar las órdenes en el módulo dormitorios.

Código sensor de luz LDR

```
valorLDR = analogRead(A0);  
if (valorLDR < 400) {  
  strldr = "FOCO APAGADO";  
  estado_mov = "encendido";  
  if (estado_mov == "encendido" && estado_movAnt != "encendido")  
    estado_movAnt = estado_mov;  
  strldr.toCharArray(valueStr, 15);  
  client.publish(LDR, valueStr, 1);  
  Serial.println("FOCO APAGADO");  
}  
} else {  
  strldr = "FOCO ENCENDIDO";  
  estado_mov = "apagado";  
  if (estado_mov == "apagado" && estado_movAnt != "apagado") {  
    estado_movAnt = estado_mov;  
    strldr.toCharArray(valueStr, 15);  
    client.publish(LDR, valueStr, 1);  
    Serial.println("FOCO ENCENDIDO");  
  }  
}
```

Figura 31. Código para confirmar si un foco está encendido o no.
Elaborado por: Juan Fonseca y Daniel Soria

En la Figura 31, se elaboró las sentencias necesarias para alertar a la aplicación web en caso de tener un foco encendido o apagado, además, el código confirma si existe o no un foco quemado en caso de encenderlo y no obtener respuesta.

Código rotación de motor

```
void input_serial() {  
    delay(1);  
    t = var.toInt();  
    if (t == 0) motor_off();  
    else {  
        if (t < 0) {  
            t = abs(t);  
            giro = true;  
        } else giro = false; }  
}
```

Figura 32. Código para rotar el motor.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 32, representa condiciones para activar las bobinas internas en sentido horario o antihorario, a su vez, en la Figura 33, con una secuencia de pasos se logró rotar el motor condicionando en un lazo repetitivo el número de vueltas que se necesite.

Código secuencia para encender y apagar el motor

```
void motor_on() {  
    for (int i = 0; i < 8; i++) {  
        int step = i;  
        if (giro) step = 8 - i;  
        digitalWrite(IN1, bitRead(halfpass[step], 0));  
        digitalWrite(IN2, bitRead(halfpass[step], 1));  
        digitalWrite(IN3, bitRead(halfpass[step], 2));  
        digitalWrite(IN4, bitRead(halfpass[step], 3));  
        delay(t); }  
}  
  
void motor_off() {  
    digitalWrite(IN1, 0);  
    digitalWrite(IN2, 0);  
    digitalWrite(IN3, 0);  
    digitalWrite(IN4, 0);  
}
```

Figura 33. Código de secuencia en ocho pasos para mover y detener las bobinas del motor.
Elaborado por: Juan Fonseca y Daniel Soria

3.2.2. Comedor - cocina.

La Figura 34, muestra el uso de una tarjeta de desarrollo ESP-8266, misma que se ubicó de manera estratégica para poder comunicar el comedor y cocina. Además, cada sector del módulo cuenta con luces independientes y, se instaló en la cocina un sensor de gas/humo MQ-6 empleado para detectar escapes de gas o alertar sobre posibles incendios.

Circuito para módulo Comedor - Cocina

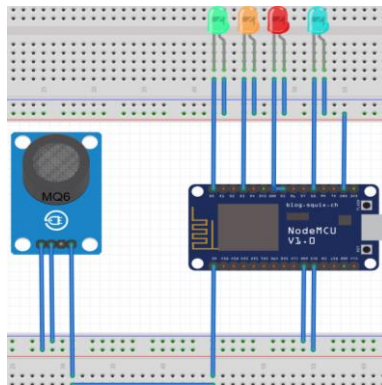


Figura 34. Diagrama del circuito para el módulo comedor y cocina con sus respectivos sensores y actuadores

Elaborado por: Juan Fonseca y Daniel Soria

3.2.2.1. Código comedor - cocina.

La Figura 35, describe el código relevante que se utilizó para configurar y ejecutar las órdenes en el módulo comedor - cocina, destacando la obtención de datos en el sensor de gas/humo MQ-6.

Código sensor de gas y humo MQ - 6

```
int var_mq6 = analogRead(A0);
if (var_mq6 >= 65) {
    float var = var_mq6;
    strmq6 = String(var, 1); //1 decimal
    strmq6.toCharArray(valueStr, 15);
    cont++;
    if (cont == 1) {
        Serial.println("Enviando: [" + String(MQ6) + "] " + strmq6);
        client.publish(MQ6, valueStr, 1);
    }
}
```

Figura 35. Código para obtención de datos en el sensor de gas/humo MQ - 6.

Elaborado por: Juan Fonseca y Daniel Soria

3.2.3. Sala - pasillo.

En la Figura 36, se utilizó una tarjeta de desarrollo ESP-32s, misma que se ubicó de manera estratégica para poder comunicar el pasillo y la sala. Además, ambos sectores del módulo cuentan con luces independientes y se dispuso también de un sensor DHT-11 para informar al usuario sobre la temperatura ambiente en caso de poseer un sistema de enfriamiento o climatización. Por otro lado, se detectó movimiento por medio del sensor PIR que, al igual que en el módulo garaje se lo activa en caso de que la vivienda se quede sola y necesite seguridad.

Circuito para módulo Sala - Pasillo

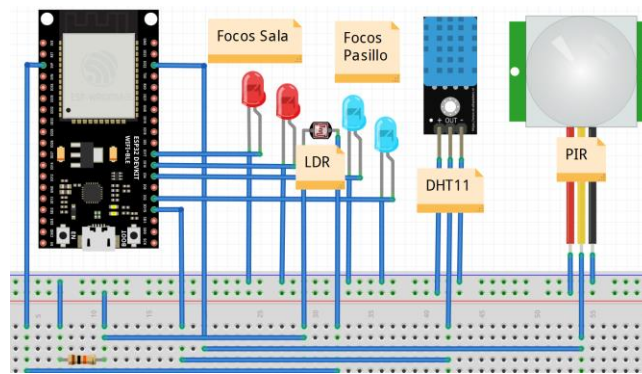


Figura 36. Diagrama del circuito para el módulo sala - pasillo con sus sensores y actuadores.
Elaborado por: Juan Fonseca y Daniel Soria

3.2.3.1. Código sala - pasillo.

La Figura 37, describe el código para medir la temperatura por medio del sensor DHT11 cada cinco segundos en el módulo sala - pasillo. Se enfatiza que para este módulo se aplicó el mismo código expuesto en la Figura 42 (módulo garaje – baño) para detectar movimiento.

Código sensor DHT 11

```
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= 5000) {
  previousMillis = currentMillis;
  float temp = dht.getTemperature();
  strtemp = String(temp, 1); //1 decimal
  strtemp.toCharArray(valueStr, 15);
  Serial.println("Enviando: [" + String(TEMPERATURA) + "] " + strtemp);
  client.publish(TEMPERATURA, valueStr, 1);
}
```

Figura 37. Código para medir temperatura con sensor DHT11 cada cinco segundos.
Elaborado por: Juan Fonseca y Daniel Soria

3.2.4. Pasillo con cámara ESP32-CAM.

La Figura 38, utilizó la placa ESP32-CAM con un adaptador USB – TTL en el pasillo para mostrar imágenes en tiempo real (streaming) mediante una comunicación inalámbrica. Además, este dispositivo se activa al detectar movimiento y captura fotografías enviándolas al usuario por medio del servicio de mensajes Telegram.

Circuito pasillo ESP32-CAM

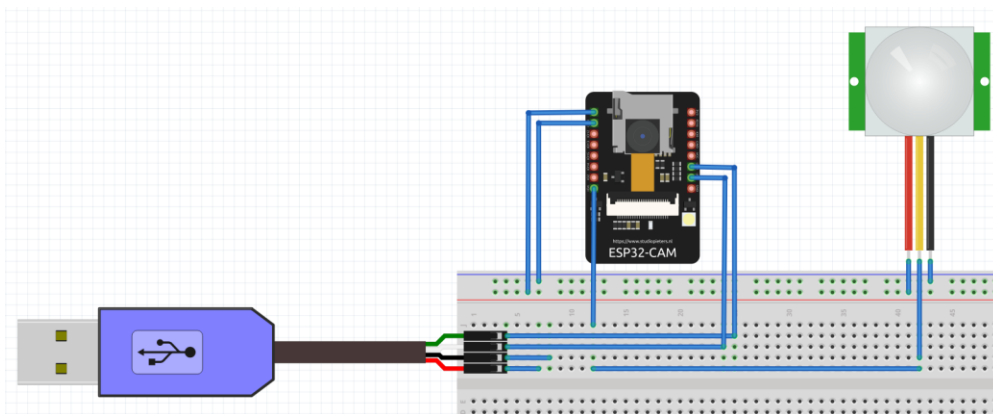


Figura 38. Diagrama del circuito para placa ESP32-CAM
Elaborado por: Juan Fonseca y Daniel Soria

3.2.4.1. Código pasillo con ESP32-CAM.

La Figura 39, describe el método para inicializar la transmisión de imágenes y video en tiempo real, utilizado para programar la placa ESP32-CAM en Arduino IDE.

Código para placa ESP32-CAM

```
void startCameraServer() {
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    config.server_port = 80;

    httpd_uri_t index_uri = {
        .uri      = "/",
        .method    = HTTP_GET,
        .handler    = stream_handler,
        .user_ctx  = NULL
    };

    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(stream_httpd, &index_uri);
    }
}
```

Figura 39. Método para inicializar la transmisión de imágenes y video en tiempo real.
Elaborado por: Juan Fonseca y Daniel Soria

Código para placa ESP32-CAM

```
pinMode(gpioPIR, INPUT_PULLUP);
int v = digitalRead(gpioPIR);
Serial.println(v);

if (mens == "on") {
    if (v == 1) {
        alerts2Telegram(token, chat_id);
        strmsg = "IMAGEN-ENVIADA";
        strmsg.toCharArray(valueStr, 15);
        client.publish(MSG, valueStr);
        delay(5000);
    } else {
        Serial.println("NO EXISTEN NOVEDADES");
    }
}
}
```

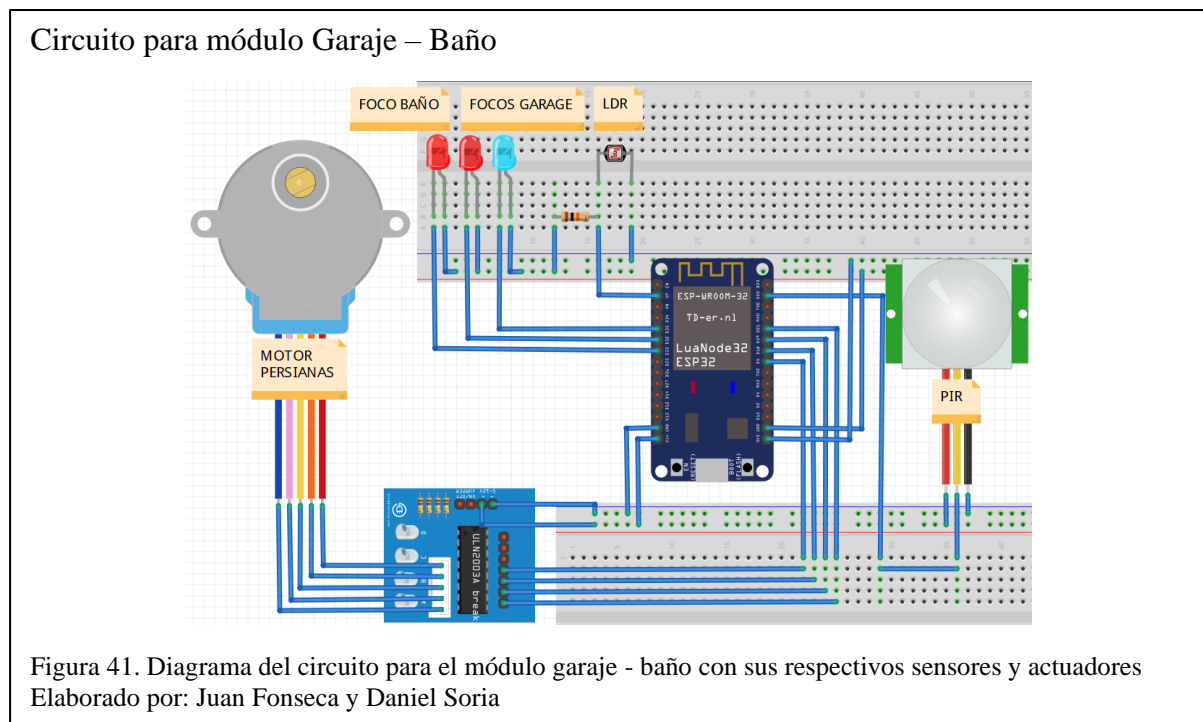
Figura 40. Procedimiento para capturar y enviar imágenes a Telegram
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 40, presenta condiciones elaboradas en la cámara para el siguiente procedimiento: en primera instancia, se alerta de un movimiento no autorizado detectado por el PIR, una vez capturado, se envía de manera instantánea al usuario una imagen en formato “.jpeg” por intermedio de la aplicación Telegram. Se enfatizó que, para utilizar este servicio el usuario debe contar con un “chat Id” y “token”, mismos que generaron un “bot” que recibió las capturas tomadas por la placa ESP32-CAM.

3.2.5. Garaje - baño.

Para este módulo se utilizó una tarjeta de desarrollo ESP-32, misma que se ubicó (Figura 41) de manera estratégica para poder comunicar el garaje y el baño, además, cuenta con luces independientes y se habilitó la función de detectar presencia por medio del sensor PIR en caso de que la vivienda se quede sola y necesite ser asegurado.

Por otro lado, se cuenta con un motor modelo 28BYJ-48 paso a paso que permitió abrir y cerrar la puerta del garaje para poder ingresar o salir con un vehículo.



3.2.5.1. Código garaje - baño.

La Figura 42, describe el código para detectar movimiento con el sensor de presencia HC-SR501 (módulo garaje – baño). Se enfatiza que, en este requerimiento se aplicó el mismo código ya expuesto en las Figuras 32 y 33 (módulo dormitorios) para encender, rotar y apagar el motor.

Código sensor PIR

```
if (digitalRead(22) == true) {  
    strPir = "Movimiento";  
    estado_mov = "intruso";  
    if (estado_mov == "intruso" && estado_movAnt != "intruso") {  
        estado_movAnt = estado_mov;  
        strPir.toCharArray(valueStr, 15);  
        client.publish(PIR, valueStr, 1);  
        Serial.println("Intruso");  
    }  
} else {  
    estado_mov = "normal";  
    if (estado_mov == "normal" && estado_movAnt != "normal") {  
        estado_movAnt = estado_mov;  
    }  
}
```

Figura 42. Código para detectar movimiento en el sensor de presencia HC-SR501
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 42, presenta las condiciones necesarias para alertar al usuario en caso de detectar movimiento, por lo tanto, si la lectura realizada en el pin digital 22 es confirmada, se publica el mensaje en el servidor MQTT.

3.3. Diagrama Lógico y Físico de Bloques de Hardware

Con todos los módulos presentados en los literales anteriores, se procedió a diseñar la estructura lógica (Figura 43) y estructura física (Figura 44) entre todas las tarjetas de desarrollo utilizadas. Para ello, se recuerda que son elementos con conexión inalámbrica Wi-Fi que facilitan su comunicación y están basados en el protocolo MQTT que, mediante la asignación de tópicos permite a un mensaje publicado ser leído por todos los participantes del mismo, siendo actualmente la opción ideal para recolección y publicación de datos.

Diagrama de bloques lógico de hardware

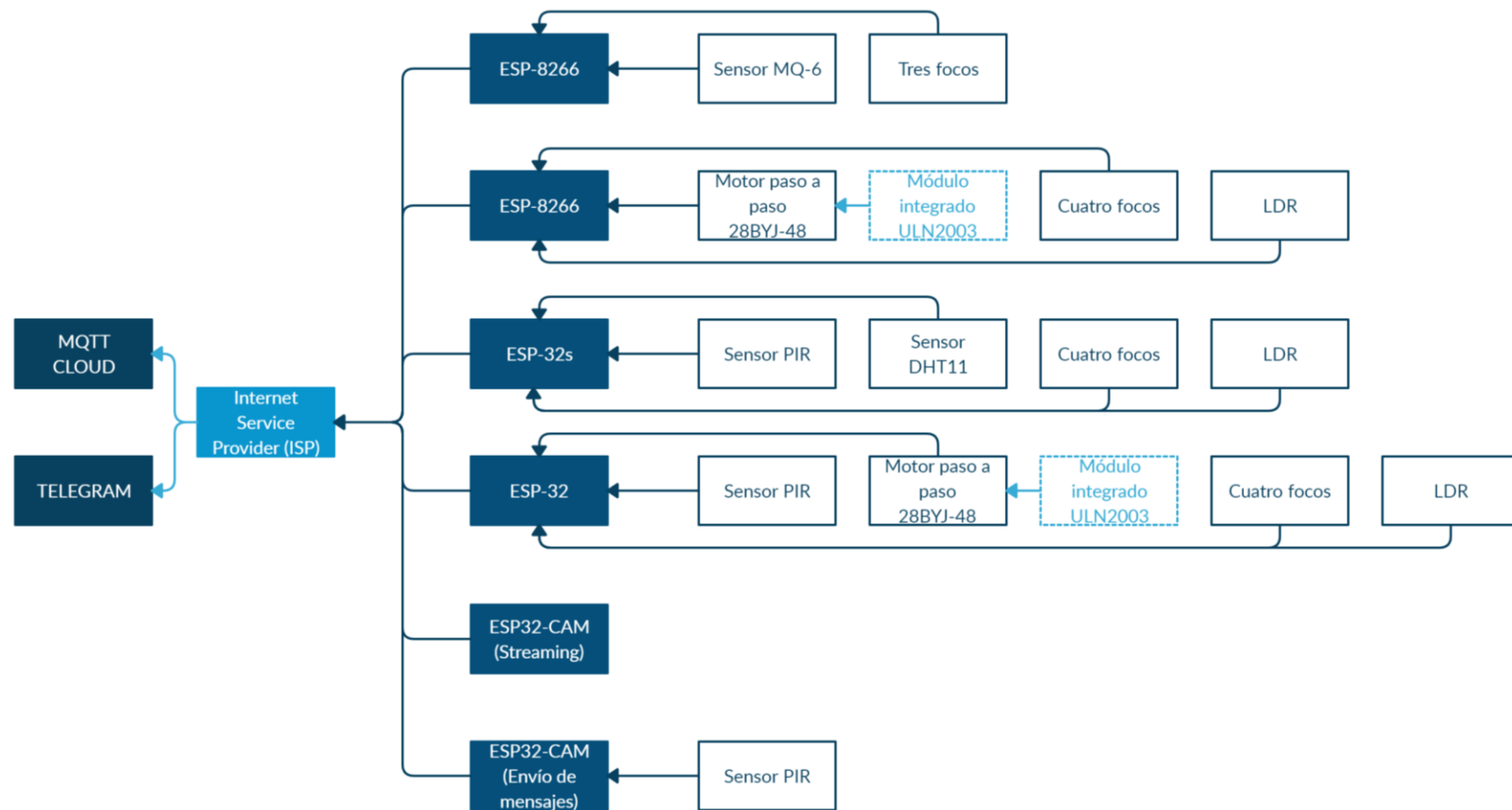


Figura 43. Diagrama lógico del circuito completo en bloques para hardware
Elaborado por: Juan Fonseca y Daniel Soria

Diagrama de bloques físico de hardware

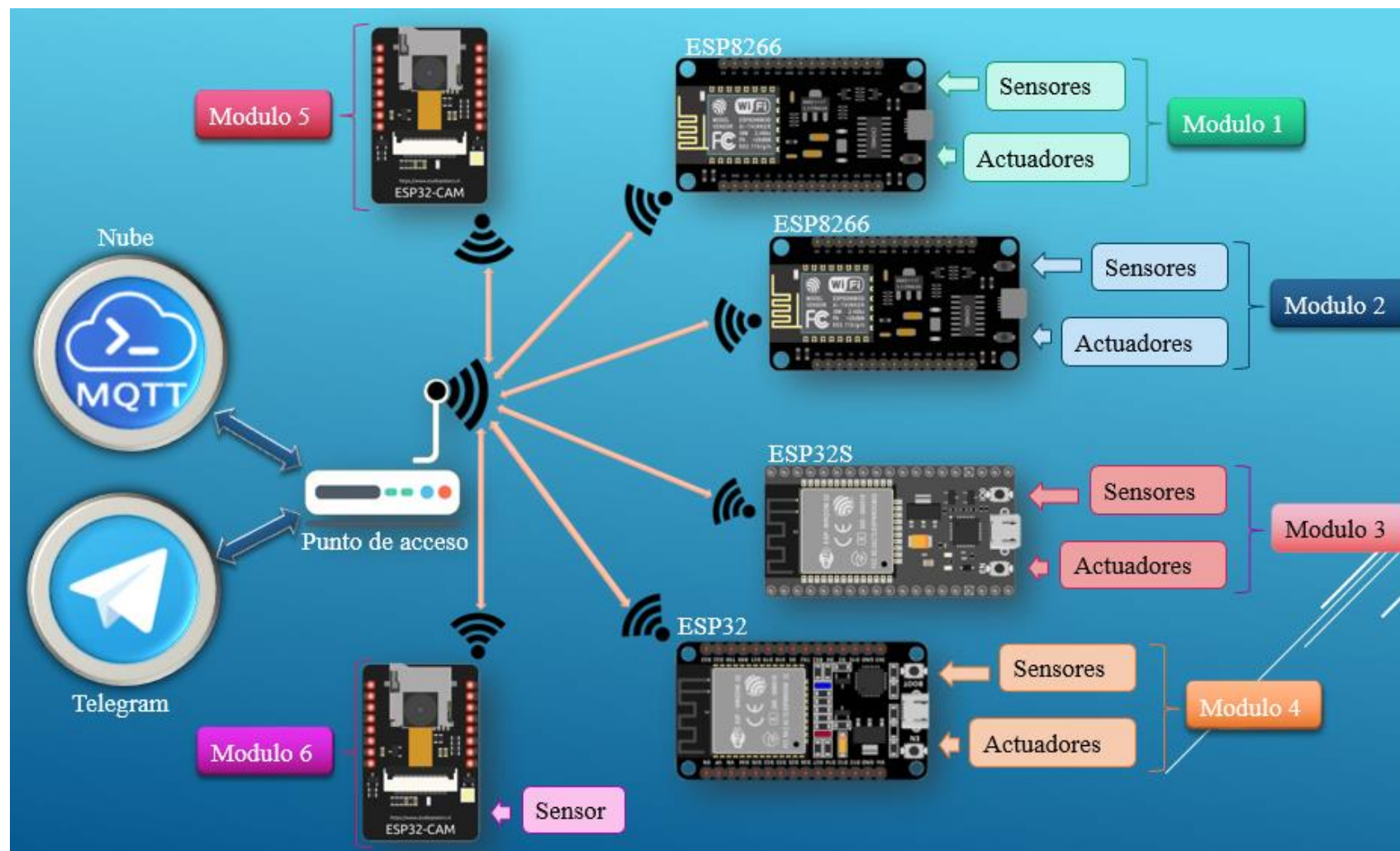


Figura 44. Diagrama físico del circuito completo en bloques para hardware
Elaborado por: Juan Fonseca y Daniel Soria

CAPÍTULO 4

4. Periodo de Pruebas

En este capítulo se ejecutaron pruebas para software y/o hardware fundamentadas en la metodología del Modelo en V del ciclo de vida (fase 5 hasta fase 7), del mismo modo, se presentaron cuadros comparativos que identifiquen el procedimiento ideal.

4.1. Pruebas de Operatividad

En este literal, se estableció un formato elaborado por el grupo de trabajo para registrar: descripción, requisitos, procedimiento y resultados de cada prueba realizada en base a las fases cinco y seis de la metodología planteada; esta información se visualizará en las Tablas 66, 68, 70, 72 y 74 según corresponda.

4.1.1. Funcionamiento de elementos.

Tabla 66. Prueba general de comunicación para placas ESP82-66 y ESP-32

PRUEBA #1	Comunicación de placas ESP con red Wi-Fi, sensores y actuadores.
DESCRIPCIÓN	Se realizó la prueba con cada una de las tarjetas de desarrollo para confirmar su funcionamiento.
REQUISITOS	Punto de acceso para comunicación inalámbrica, librerías ESP, Arduino IDE, cables
PROCEDIMIENTO	Configuración inicial de placas ESP en Arduino IDE. Ingreso de usuario (SSID) y contraseña para red Wi-Fi. Instalación de librerías necesarias en placas ESP. Interacción para placas ESP con sensores y actuadores. Visualización de datos en monitor serie.
RESULTADO	Se configuró cada placa ESP obteniendo comunicación exitosa con internet, sensores y actuadores de acuerdo a su función.

Nota: Esta tabla contiene en detalle lo necesario para ejecutar la prueba con dispositivos en general.

Elaborado por: Juan Fonseca y Daniel Soria

4.1.2. Placa ESP y MySQL, php, get/post, Arduino IDE.

Tabla 67. Tiempos de respuestas placa ESP con HTTP, MySQL, get/post, Arduino IDE

PROTOCOLO HTTP, MYSQL, GET/POST, ARDUINO IDE	
NÚMERO DE PRUEBA	TIEMPO [s]
1	2,108
2	1,571
3	1,583
4	1,768
5	1,807
6	2,095
7	1,984
8	1,870
9	2,233
10	2,055
TOTAL	19,074
PROMEDIO	1,91[s]

Nota: Esta tabla contiene tiempo y promedio en segundos para ejecutar la prueba del protocolo HTTP.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 67, presenta los tiempos registrados con un cronómetro para 10 pruebas de comunicación entre los elementos mencionados con un promedio de 1,91 segundos de retardo en su respuesta; además, se resalta que tanto estas pruebas como las siguientes fueron ejecutadas con un “ISP” que cuenta con tecnología de fibra óptica y una velocidad de transmisión de 100 Mbps.

Por consiguiente, se realizó una adición para la columna TIEMPO obteniendo un valor denominado “TOTAL”, no obstante, en la fila “PROMEDIO” se dividió el “TOTAL” para el número de pruebas, alcanzando el tiempo promedio de respuesta con 2 decimales en el protocolo mencionado.

Este procedimiento fue repetido para las Tablas 69, 71 y 73 con sus datos respectivos.

Tabla 68. Prueba de funcionamiento para placa ESP con MySQL, PHP, get/post y Arduino IDE

PRUEBA #2	Insertar y extraer datos con método GET/POST, MySQL, PHP y Arduino IDE
DESCRIPCIÓN	Se realizó la prueba con un sensor de temperatura/humedad. Este procedimiento sirvió de manera similar para los demás sensores y actuadores del proyecto técnico.
REQUISITOS	Cliente y librerías ESP, script PHP, base de datos MySQL, Arduino IDE.
PROCEDIMIENTO	<p>Disponibilidad de un hosting que ofrezca PHP y MySQL.</p> <p>Creación de base de datos, script PHP con métodos GET y POST para enviar y recibir datos.</p> <p>Instalación librerías HTTPClient, WiFiClient, WiFi.</p> <p>Interacción para placa ESP con herramientas antedichas.</p> <p>Muestra y comunicación de datos resultantes.</p>
RESULTADO	Se obtuvo una conexión exitosa a la base de datos donde se registró y se consultó los valores de la placa ESP, sin embargo, existió un retardo promedio de 1,91 segundos en cada petición al servidor.

Nota: Esta tabla contiene en detalle lo necesario para ejecutar la prueba del protocolo HTTP.

Elaborado por: Juan Fonseca y Daniel Soria

Monitor serie con información de placa ESP con protocolo HTTP

```

WiFi connected
LOCAL IP: 192.168.100.15

0
Led Apagado
No motion detected.
Código HTTP ► 200
El servidor respondió ▼
Datos ingresados correctamente.</br><a href= formulario.html >Regresar estado</a>

assertion "heap != NULL && "free() target pointer is outside heap areas"" failed:
<
☐ Autoscroll ☐ Mostrar marca temporal

```

Figura 45. Contiene el resultado de la tabla 68.

Elaborado por: Juan Fonseca y Daniel Soria

La Figura 45, contiene la información del comportamiento de la placa ESP conjuntamente a sensores y actuadores bajo el protocolo HTTP, resultado de la Tabla 68.

4.1.3. Placa ESP y WebSockets, php, Arduino IDE.

Tabla 69. Tiempos de respuestas placa ESP con WebSockets, PHP, Arduino IDE

PROTOCOLO WebSockets, PHP, ARDUINO IDE	
NÚMERO DE PRUEBA	TIEMPO [s]
1	0,786
2	0,886
3	0,770
4	0,808
5	0,918
6	0,953
7	1,010
8	0,860
9	0,952
10	0,957
TOTAL	8,9
PROMEDIO	0,89 [s]

Nota: Esta tabla contiene tiempo y promedio en segundos para ejecutar la prueba del protocolo WebSockets.
Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 69, presenta los tiempos registrados con un cronómetro para 10 pruebas de comunicación entre los elementos mencionados con un promedio de 0,89 segundos de retardo en su respuesta. Se aplicó el mismo procedimiento que la Tabla 67 para calcular el promedio.

Tabla 70. Prueba de funcionamiento para placa ESP con WebSockets, PHP y Arduino IDE.

PRUEBA #3	Interacción de datos con WebSockets, PHP y Arduino IDE
DESCRIPCIÓN	Se realizó la prueba con un sensor de movimiento. Este procedimiento sirvió de manera similar para los demás sensores y actuadores del proyecto técnico.
REQUISITOS	Cliente y librerías ESP, servidor PHP, Arduino IDE, navegador web.
PROCEDIMIENTO	<p>Servidor Apache XAMPP con puerto 8080 habilitado.</p> <p>Creación de scripts para servidor PHP a fin de escuchar peticiones de todos los clientes conectados a la red.</p> <p>Instalación librerías Arduino, WebSockets, WiFi.</p> <p>Interacción para placa ESP con herramientas antedichas.</p> <p>Muestra y comunicación de datos resultantes.</p>
RESULTADO	Se obtuvo una conexión exitosa al servidor PHP de WebSockets donde se comunicó la placa ESP con el navegador web, sin embargo, este método no fue idóneo para la transmisión y recepción de datos con múltiples dispositivos porque generó cuellos de botella en la red y un retardo promedio de 0,89 segundos de respuesta.

Nota: Esta tabla contiene en detalle lo necesario para ejecutar la prueba del protocolo WebSockets.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 46, muestra el envío y recepción de datos bajo el protocolo WebSockets que muestra la comunicación entre dos clientes y el servidor, resultado de la Tabla 70.

Envío y recepción de datos bajo el protocolo WebSockets

[illegible]

Figura 46. Contiene el resultado de la tabla 70.
Elaborado por: Juan Fonseca y Daniel Soria

4.1.4. Placa ESP y MySQL, Arduino IDE.

Tabla 71. Tiempos de respuestas placa ESP con MySQL, Arduino IDE

MySQL connector, Arduino IDE	
NÚMERO DE PRUEBA	TIEMPO [s]
1	1,819
2	1,574
3	1,325
4	1,588
5	1,677
6	1,421
7	1,590
8	1,438
9	1,621
10	1,552
TOTAL	15,605
PROMEDIO	1,56 [s]

Nota: Esta tabla contiene tiempo y promedio en segundos para ejecutar la prueba con MySQL.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 71, presenta los tiempos registrados con un cronómetro para 10 pruebas de comunicación entre los elementos mencionados con un promedio de 1,56 segundos de retardo en su respuesta. Se aplicó el mismo procedimiento que la Tabla 67 para calcular el promedio.

Tabla 72. Prueba para funcionamiento de placa ESP con MySQL y Arduino IDE

PRUEBA #4	Insertar y extraer datos con MySQL y Arduino IDE
DESCRIPCIÓN	Se realizó la prueba con un sensor de gas y humo MQ-6. Este procedimiento sirvió de manera similar para los demás sensores y actuadores del proyecto técnico.
REQUISITOS	Cliente y librerías ESP, MySQL, Arduino IDE.
PROCEDIMIENTO	Disponibilidad de un hosting que ofrezca MySQL. Creación de base de datos. Instalación librerías MySQL_Connection, MySQL_Cursor, WiFi. Interacción para placa ESP con herramientas antedichas. Muestra y comunicación de datos resultantes.
RESULTADO	Se obtuvo una conexión exitosa a la base de datos donde se registró y se consultó los valores de la placa ESP, sin embargo, existió un retardo promedio de 1,56 segundos de respuesta en cada petición al servidor.

Nota: Esta tabla contiene en detalle lo necesario para ejecutar la prueba con MySQL.

Elaborado por: Juan Fonseca y Daniel Soria

La Figura 47, muestra el envío y recepción de datos con MySQL que muestra la comunicación bajo el modelo cliente - servidor, resultado de la Tabla 72.

Envío y recepción de datos con MySQL

```

Disconnected.
PRUEBA SERVIDOR EXTERNO = 2 cm
Connected to server version 5.6.40-84.0-log
Recording data.
INSERT INTO mykocomun_prueba.datos(chipId, fecha, estado_sensor) VALUES ('782af9bf713c', CURRENT_TIMESTAMP, 3);
Disconnected.
Registro Grabado

Connecting MySQL...
Connected to server version 5.6.40-84.0-log
Conectado a MySQL
Connected to server version 5.6.40-84.0-log
Disconnected.
PRUEBA SERVIDOR EXTERNO = 3 cm
Connected to server version 5.6.40-84.0-log
  
```

☒ Autoscroll
 ☐ Mostrar marca temporal
 Sin ajuste de línea
 115200 baudio

Figura 47. Contiene el resultado de la tabla 72.
Elaborado por: Juan Fonseca y Daniel Soria

4.1.5. Placa ESP y MQTT-SN, Arduino IDE.

Tabla 73. Tiempos de respuestas placa ESP con MQTT-SN, Arduino IDE

PROTOCOLO MQTT-SN, Arduino IDE	
NÚMERO DE PRUEBA	TIEMPO [s]
1	0,366
2	0,511
3	0,492
4	0,505
5	0,478
6	0,466
7	0,604
8	0,392
9	0,497
10	0,455
TOTAL	4,766
PROMEDIO	0,48 [s]

Nota: Esta tabla contiene tiempo y promedio en segundos para ejecutar la prueba del protocolo MQTT-SN.
Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 73, presenta los tiempos registrados con un cronómetro para 10 pruebas de comunicación entre los elementos mencionados con un promedio de 0,48 segundos de retardo en su respuesta. Se aplicó el mismo procedimiento que la Tabla 67 para calcular el promedio.

Tabla 74. Prueba de funcionamiento para placa ESP con MQTT-SN y Arduino IDE

PRUEBA #5	Interacción de datos con MQTT-SN y Arduino IDE
DESCRIPCIÓN	Se realizó la prueba con un sensor de gas y humo MQ-6. Este procedimiento sirvió de manera similar para los demás sensores y actuadores del proyecto técnico.
REQUISITOS	Cliente y librerías ESP, CloudMQTT, Arduino IDE, navegador web.
PROCEDIMIENTO	Disponer de una cuenta de usuario en la nube MQTT. Instalación librerías PubSubClient, WiFi. Suscripción de topics del cliente en el servidor MQTT. Interacción de placa ESP con herramientas antedichas. Visualización e interacción de datos resultantes.
RESULTADO	Se obtuvo una conexión exitosa al servidor MQTT donde se registró y se consultó los valores de las placas ESP teniendo un retardo de 0,48 segundos de respuesta dadas las pruebas de experimentación. Además, actualmente es el protocolo ideal para la relación entre IoT y Sistemas Embebidos.

Nota: Esta tabla contiene en detalle lo necesario para ejecutar la prueba con protocolo MQTT-SN.

Elaborado por: Juan Fonseca y Daniel Soria

La Figura 48, muestra el envío y recepción de datos bajo protocolo MQTT que muestra la comunicación entre cliente y servidor en la interfaz de MQTT-BROKER, donde se enciende y se apaga un foco implementando hardware con software, resultado de la Tabla 74.

Envío y recepción de datos bajo protocolo MQTT

The image shows two side-by-side windows. The left window is a web-based MQTT console (https://api.cloudmqtt.com/console/02660587/websocket). The right window is an Arduino IDE showing a C++ program for an ESP8266 that communicates via MQTT.

MQTT Console (Left Window):

- Topic:** /controlador/foco6
- Message:** ON
- Send:** [Button]
- Clear session:** Clear any data for a client id. Client ID: [Input field]. Clear: [Button]
- Received messages:**

Topic	Message
/controlador/con	P1-ESP8266 UP
/controlador/ldr	FOCO APAGADO
/controlador/persiana	A
/controlador/persiana	C
/controlador/persiana	S
/controlador/foco6	ON
/controlador/ldr	FOCO ENCENDIDO

Arduino IDE (Right Window):

```

..Intentando conexion MQTT...conectado
FOCO APAGADO
Mensaje Recibido: [/controlador/persiana] A
ABRIENDO PERSIANA
Mensaje Recibido: [/controlador/persiana] C
CERRANDO PERSIANA
Mensaje Recibido: [/controlador/persiana] S
PERSIANA DETENIDA
Mensaje Recibido: [/controlador/foco6] ON
FOCO ENCENDIDO
FOCO APAGADO
FOCO ENCENDIDO
FOCO APAGADO
FOCO ENCENDIDO
FOCO APAGADO
FOCO ENCENDIDO
FOCO APAGADO

//-----MOTOR-----
if (var == "1" || var == "-1" || var == "0") input_serial(); // recoge el valor asignado a var, despues de una orden del servidor
if (t != 0) motor_on(); //Cuando t es diferente de 0, el motor el motor esta encendido.

//-----LDR-----
valorLDR = analogRead(A0);
//Serial.println(valorLDR);
if (valorLDR < 400) {
  strldr = "FOCO ENCENDIDO";
  estado_mov = "encendido";
  if (estado_mov == "encendido" && estado_movAnt != "encendido") {
    //Serial.println(estado_mov);
    estado_movAnt = estado_mov;
    strldr.toCharArray(valueStr, 15);
    client.publish(LDR, valueStr);
    Serial.println("FOCO ENCENDIDO");
  }
} else {
  estado_mov = "apagado";
  strldr = "FOCO APAGADO";
  if (estado_mov == "apagado" && estado_movAnt != "apagado") {
    //Serial.println(estado_mov);
    estado_movAnt = estado_mov;
    strldr.toCharArray(valueStr, 15);
    client.publish(LDR, valueStr);
    Serial.println("FOCO APAGADO");
  }
}
  
```

Figura 48. Contiene el resultado de la tabla 74.
Elaborado por: Juan Fonseca y Daniel Soria

4.2. Cuadros Comparativos de Resultados

Una vez realizadas las pruebas con todos los requerimientos solicitados, se verificó la efectividad de cada opción ejecutada mediante parámetros establecidos por el grupo de trabajo publicados en la Tabla 76.

Por consiguiente, se tomó como referencia uno de los protocolos domóticos de comunicación conocido como “C-Bus” presentado en la Figura 49, mismo que se enfoca principalmente en instalaciones del hogar, posee una arquitectura similar al modelo “OSI” de siete capas y es el único protocolo que garantiza una conexión inalámbrica a la red por medio del estándar WiFi 802,11. Además, C-Bus y KNX alcanzan una velocidad de transmisión de hasta 1200 Bps (Ruiz Buitrón, 2020).

Protocolos de comunicación domótica más utilizados			
Protocolo	Velocidad de transmisión	Distancias de alcance	Medios de transmisión utilizados
X 10	50 bps a 60 bps	No descrito	Cable eléctrico
KNX	150 Bps a 1200 Bps	300 m a 1000 m	Par trenzado, cable eléctrico, RF, IR y Ethernet
ZigBee	2,5 KBps a 32 KBps	10 m a 75 m	Radiofrecuencia (RF)
LonWorks	200 Bps a 160 KBps	1500 m a 2700 m	Par trenzado, cable eléctrico, cable coaxial y RF
Z-Wave	1,2 KBps a 5 KBps	1 m a 75 m	RF
C-Bus	150 Bps a 1200 Bps	1000 m	Par trenzado, cable eléctrico, RF, IR, Ethernet y wifi

Figura 49. Protocolos domóticos utilizados para la comunicación especificando su velocidad de transmisión, distancia de alcance y medio de transmisión.
Fuente: (Ruiz Buitrón, 2020)

Por esta razón, en la Tabla 75, se asignó un valor numérico de uno a cinco, equivalente a la categoría según el comportamiento de cada protocolo probado y fundamentado en la información investigada.

Tabla 75. Rúbrica de evaluación

Categoría	Valor	Observación
Sobresaliente	5	Cumple con el objetivo plenamente.
Notable	4	Cumple con el objetivo, pero con latencia.
Aprobado	3	Cumple con el objetivo aceptablemente.
Deficiente	2	Cumple insatisfactoriamente con el objetivo.
Pésimo	1	No cumple.

Nota: Esta tabla contiene criterios acordados por el grupo de trabajo para calificar.

Elaborado por: Juan Fonseca y Daniel Soria

Se compararon los elementos desde la Tabla 67 hasta la Tabla 74, para evaluar eficacia en cada procedimiento representados en la Tabla 76.

Tabla 76. Cuadro comparativo entre pruebas realizadas

Parámetro	HTTP, MySQL	MySQL	WebSockets	MQTT-SN
	MAGNITUDES SIN UNIDAD			
Compatibilidad entre dispositivos y protocolos de comunicación	4	4	5	5
Integración exitosa entre sensores, actuadores y placas de desarrollo	4	4	4	5
Tiempo de respuesta	2	3	5	5
Calidad de servicio (QoS) para la comunicación	5	5	5	5
Integración de uno o más clientes al protocolo utilizado	3	3	3	4
TOTAL	18	19	22	24
PORCENTAJE OBTENIDO	72 %	76 %	88 %	96 %

Nota: Esta tabla contiene la evaluación de elementos según criterios del equipo de trabajo.

Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 76, indica hitos con mayor impacto en el proyecto técnico, así mismo, se calificó según la Tabla 75, cada prueba de acuerdo al parámetro seleccionado; por consiguiente, se realizó una adición por columna a las calificaciones para obtener un valor denominado “TOTAL”.

No obstante, en la fila “PORCENTAJE OBTENIDO” se dividió el “TOTAL” para el número de parámetros, de manera que, mediante una regla de tres directa se multiplicó ese valor por el 100% y finalmente se dividió ese resultado para 25 (sumatoria equivalente a la máxima puntuación de 5 por cada parámetro según la Tabla 75) alcanzando el resultado final, teniendo en cuenta que se investigó varias opciones hasta acercarse lo más posible al 100%, siendo en este caso la mejor opción el protocolo MQTT-SN con el **96 %** de efectividad.

Gráfico de barras para comparación de protocolos

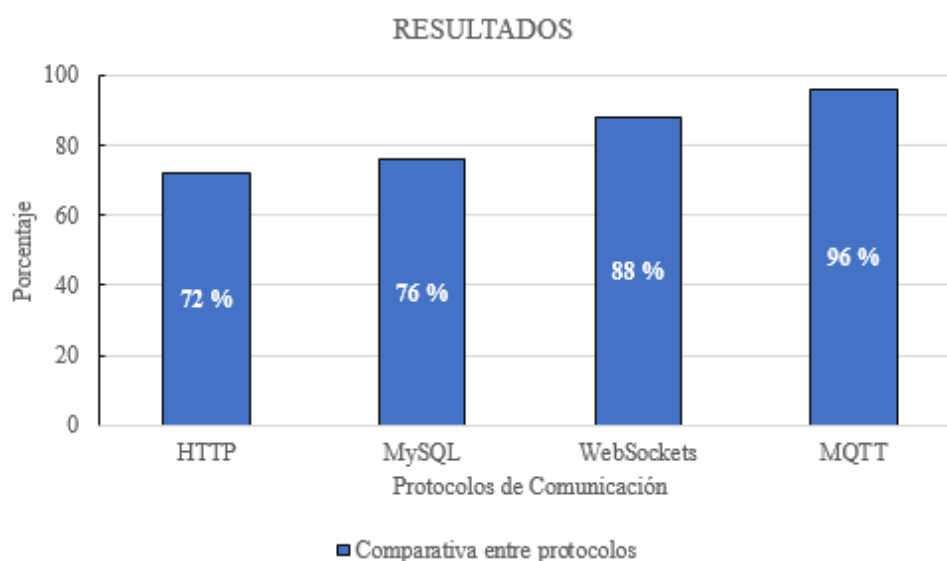


Figura 50. Representación en barras para comparación de protocolos de comunicación.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 50, representa por medio de un gráfico de columnas apiladas los resultados que se obtuvieron en la Tabla 76, reflejando al protocolo MQTT-SN como la opción que garantizó el funcionamiento correcto de este proyecto técnico.

Tabla 77. Evaluación del producto final

Parámetro	Valor	Observación
Compatibilidad con IoT y Sistemas Embebidos	5	Es el mejor protocolo de acuerdo a las necesidades del producto.
Precio de los recursos utilizados	5	Es barato a comparación de otros servicios tecnológicos.
Envío y recepción de datos altamente fiables	5	No se pierde la información bajo el modelo TCP/IP.
Ahorro de consumo energético	5	Alertas en caso del uso innecesario de elementos.
Brinda seguridad al entorno físico	5	Transmisión en tiempo real en caso de detectar presencia no permitida por el usuario.
Velocidad de respuesta entre dispositivos y servidor	5	Respuesta inmediata a las peticiones del usuario.
Nivel de interacción del usuario con la aplicación	4	El usuario final debe tener conocimientos básicos para el uso de un navegador web.
TOTAL	34	
PORCENTAJE OBTENIDO	97,14 %	

Nota: Esta tabla evaluará el protocolo MQTT-SN como opción final.
Elaborado por: Juan Fonseca y Daniel Soria

La Tabla 77, indica hitos con mayor impacto para esta alternativa, por ende, se ratificó la eficacia del protocolo MQTT-SN utilizando el mismo procedimiento aplicado en la Tabla 76 con un 97,14% de efectividad; además, se argumentó en la columna “Observación” cada parámetro calificado.

4.3. Confiabilidad de Datos Enviados y Recibidos con Protocolo MQTT-SN

Se procedió a verificar la confiabilidad de los datos enviados y recibidos una vez ya seleccionado el protocolo adecuado. Para tal efecto, se argumentó la calidad de servicio (QoS) que ofrece MQTT-SN con referencia a sus mecanismos de confiabilidad.

Hervas Parra (2018) afirma, que existen dos niveles de calidad de servicio establecidos por el protocolo MQTT para el intercambio de paquetes (Hervas Parra, 2018, pág. 44). Estos son:

- Nivel 0: Brinda un servicio con mejor esfuerzo para su entrega, es decir, el mensaje enviado por el cliente es único y no comprueba si en verdad alcanzó su destino. No obstante, los mensajes con gran tamaño, probablemente se pueden perder al existir una red inalámbrica, algo que es muy común en este tipo de ambientes (Hervas Parra, 2018, pág. 44).
- Nivel 1: “Los mensajes son reenviados por el cliente hasta que sean confirmados por el receptor, sin embargo, este nivel de calidad de servicio permite que los paquetes alcancen su destino en varias ocasiones a causa de las retransmisiones” (Hervas Parra, 2018, pág. 44).

Por otra parte, MQTT Broker utiliza la tecnología SSL/TLS para la protección de los datos enviados y recibidos, es decir, se establece una comunicación segura TCP/IP entre cliente – servidor y viceversa. Utilizando esta información investigada, se definió lo siguiente para verificar la confiabilidad:

- Se aplicó “QoS nivel 1” a cada cliente que publica una orden o un estado para un sensor o actuador y fue recibida por el servidor MQTT broker.

Función publish ()

```
client.publish(PIR, valueStr, 1); client.publish(LDR, valueStrLDR, 1);
```

Figura 51. Función publish () para envío de datos.

Elaborado por: Juan Fonseca y Daniel Soria

La Figura 51, muestra el mensaje de publicación por parte del cliente para dos diferentes sensores; además, la estructura de esta función es: nombre del “topic”, estado de elemento, nivel QoS.

- Se diseñó e implementó en cada placa el método “callback ()”, mismo que descifra un mensaje enviado desde el servidor MQTT Broker y a su vez, ordena a la placa recibir y ejecutar acciones como encender o apagar focos, abrir o cerrar puerta y/o persiana.

Método callback ()

```
void callback(char* topic, byte* payload, unsigned int length) {  
    char PAYLOAD[5] = "    ";  
    Serial.print("Mensaje Recibido: [");  
    Serial.print(topic);  
    Serial.print("] ");  
    for (int i = 0; i < length; i++) {  
        PAYLOAD[i] = (char)payload[i];  
    }  
    Serial.println(PAYLOAD);  
    if (String(topic) == String(FOCO1)) {  
        if (payload[1] == 'N') {  
            digitalWrite(2, HIGH);  
        }  
    }  
}
```

Figura 52. Método callback () para descifrar datos recibidos en el cliente.

Elaborado por: Juan Fonseca y Daniel Soria

En la Figura 52, se reciben los parámetros: nombre del “topic” y orden enviada por el servidor; entonces, el cliente descifra este dato en caso de poseer QoS nivel 1 en la comunicación.

En resumen, los datos del proyecto cuentan con el mecanismo de confiabilidad de calidad de servicio nivel uno, confirmando si un paquete fue enviado y entregado y viceversa.

4.4. Optimización de Dispositivos

A fin de comprobar la eficiencia en los dispositivos seleccionados, se comparó información investigada y presentada para cada elemento en la Tabla 1 y Tabla 10. Se concluyó que existían limitantes al utilizar un solo modelo de placa, mismos que fueron comprobados y expuestos a continuación:

- **Limitante:** En primer lugar, se utilizó placas ESP-8266 que poseen un solo núcleo para procesar algoritmos, además, incluye una única entrada de lectura analógica (A0), limitando el uso de sensores con entrada o salida analógica a uno.
- **Optimización:** Para mejorar el procesamiento de algoritmos, se reemplazó las placas ESP-8266 por placas ESP-32, dispositivos que poseen dos núcleos y trabajan en paralelo, es decir, realizan dos tareas al mismo tiempo. De este modo, se investigó e implementó la clase “TaskHandle_t()” (Herranz, 2019) en estas placas, activando el segundo núcleo e incrementando su capacidad de procesos. Por otra parte, las placas ESP-32 disponen de 16 pines analógicos de entrada y salida (Espressif Inc, 2019a), permitiendo el uso de uno o más sensores analógicos en una misma placa.

Una vez que se sustituyó las placas, se presentó el impedimento para procesar todas las órdenes solicitadas en cada algoritmo, no obstante, surgió lo siguiente:

- **Limitante:** El reinicio de las placas ESP-32 se volvió frecuente, se analizó y se confirmó que el motivo era la carga excesiva de código debido a su memoria interna limitada; en consecuencia, cada algoritmo desarrollado colapsó el procesamiento de las placas, puesto que sus procesos excedieron la capacidad de memoria, reiniciando los dispositivos de forma automática.
- **Optimización:** Se reestructuró el código desarrollado aplicando programación orientada a objetos, estableciendo funciones, clases y/o métodos para ejecutar procesos que utilicen

únicamente los recursos necesarios. De este modo, se suprimió en su totalidad el reinicio automático de los dispositivos.

Por consiguiente, al mejorar el rendimiento de los elementos utilizados, se buscó una manera de mejorar el confort dentro del ambiente domiciliario para el usuario final, estableciendo que:

- Las placas ESP-8266 y ESP-32 poseen una comunicación no guiada o inalámbrica para enviar y recibir datos a través de señales microondas sin que el usuario lo note.
- Para implementar este producto no es necesario adecuar o modificar el espacio físico del hogar mediante una obra civil, porque son dispositivos independientes con voltaje de 5 [V]; además, sus dimensiones (Tabla 1 y Tabla 10) son relativamente pequeñas y el usuario no notará su instalación física.
- Se configuró pines en las placas para aumentar focos en cualquier módulo del hogar si así lo requiere, es decir, el usuario no interactúa con los dispositivos implementados.
- Inicialmente, la cámara implementada enviaba capturas de imagen a Google Drive, sin embargo, se evidenció una continuidad inestable en el servicio prestado por la nube. Por ende, se investigó y desarrolló un algoritmo que permitió enviar alertas al usuario desde la ESP32-CAM hacia la interfaz web por medio de una notificación, y a través de la aplicación Telegram, enviando un mensaje instantáneo al teléfono del usuario, proceso explicado en el Capítulo 3, Figura 40.

4.5. Análisis de Consumo Energético para Focos del Hogar

En este literal, se analizó y comparó el consumo energético generado por focos dentro de un hogar, por esta razón, se realizaron dos tipos de pruebas:

- **PRUEBA A:** Encendido y apagado de focos sin intervención del grupo de trabajo.
- **PRUEBA B:** Encendido y apagado de focos con la intervención del grupo de trabajo, por lo tanto, ambos autores de este proyecto técnico simulaban la función de alertar al usuario en caso de poseer focos encendidos cuando no lo requirió.

El cálculo para el consumo de energía eléctrica mostrado en la Figura 53, se determinó con los siguientes requisitos: potencia de focos y número de horas diarias utilizado en el mes. A continuación, se define la fórmula (CENTROSUR, 2018):

Fórmula para consumo energético

$$CONSUMO = \frac{P * horas\ diarias * días}{1000}$$

Figura 53. Fórmula para calcular consumo de energía eléctrica para un foco
Fuente: (CENTROSUR, 2018)

Donde:

- **P:** Potencia individual de foco [W]
- **horas diarias:** número de horas de uso diario [horas]
- **días:** Número de días [días]
- División para 1000 con el fin de obtener resultados en KWh.

Pruebas registradas durante un mes

PRUEBA A		PRUEBA B	
NÚMERO DE DÍAS	HORAS (h)	NÚMERO DE DÍAS	HORAS (h)
1	6	1	4
2	5	2	2
3	7	3	5
4	4	4	4
5	4	5	3
6	6	6	4
7	7	7	2
8	6	8	4
9	8	9	4
10	6	10	3
11	7	11	3
12	6	12	4
13	4	13	3
14	8	14	4
15	6	15	2
PROMEDIO:	6,00	PROMEDIO:	3,40

Figura 54. Pruebas ejecutadas en un mismo escenario durante 30 días.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 54, presenta dos muestras con información sobre el consumo energético, estas son: “PRUEBA TIPO A” y “PRUEBA TIPO B”, donde se calculó el promedio del número de horas consumidas al día durante el periodo de un mes, de acuerdo a las condiciones ya explicadas al inicio de este literal. De este modo, el escenario utilizado disponía de 4 focos ahorradores distribuidos en diferentes sectores, imitando a escala real lo planteado en este proyecto técnico.

Para asignar el valor de 0,0951 [USD] en la variable “PRECIO POR KILOWATT” de la Figura 55, se accedió e investigó en la página oficial de la Agencia de Regulación y Control de Electricidad, aseverando que es el precio oficial cobrado por la Empresa Eléctrica Quito. (ARCONEL, 2017).

De la misma manera, se investigó la potencia para cada tipo de foco utilizado en una residencia, estos son: incandescentes, ahorradores y led (Ulloa Arízaga, 2015, pág. 151).

Entonces, se procedió a calcular los tipos de consumo en potencia y en dólares basándose en la “PRUEBA A” con un promedio de 6 horas diarias perteneciente a la Figura 54, es decir, sin la intervención del grupo de trabajo.

Variables para consumo energético

VARIABLES	
NÚMERO DE DÍAS AL AÑO:	365
NÚMERO DE DÍAS AL MES:	15
NÚMERO DE HORAS AL DÍA:	6
NÚMERO DE FOCOS:	4
COSTO USD/KWh	0,0951

Figura 55. Variables para cálculo de consumo de energía eléctrica en prueba A.
Elaborado por: Juan Fonseca y Daniel Soria

Consumo energético diario para un foco

CÁLCULO DE CONSUMO DE FOCOS			
TIPO DE FOCO	POTENCIA	CONSUMO AL DÍA (w)	Kwh/DÍA
INCANDESCENTE	60	360	0,36
AHORRADOR	20	120	0,12
LED	8	48	0,048

Figura 56. Cálculo de consumo diario de energía para un foco
Elaborado por: Juan Fonseca y Daniel Soria

Consumo energético mensual para un foco

CÁLCULO DE CONSUMO DE FOCOS			
TIPO DE FOCO	POTENCIA	CONSUMO AL MES(w)	Kwh/MES
INCANDESCENTE	60	5400	5,4
AHORRADOR	20	1800	1,8
LED	8	720	0,72

Figura 57. Cálculo de consumo mensual de energía para un foco

Elaborado por: Juan Fonseca y Daniel Soria

Consumo energético anual para un foco

CÁLCULO DE CONSUMO DE FOCOS			
TIPO DE FOCO	POTENCIA	CONSUMO AL AÑO (w)	Kwh/AÑO
INCANDESCENTE	60	131400	131,4
AHORRADOR	20	43800	43,8
LED	8	17520	17,52

Figura 58. Cálculo de consumo anual de energía para un foco

Elaborado por: Juan Fonseca y Daniel Soria

Las Figuras 56, 57 y 58, exponen las tablas con sus cálculos respectivos para obtener el consumo de energía diaria, mensual y anual con un foco, especificando el tipo y potencia del mismo; por lo tanto, se utilizó la ecuación presentada en la Figura 53.

Costos de uno y “n” focos en un día, 15 días y un año

TIPO DE FOCO	AL DÍA [USD]		AL MES [USD]		AL AÑO [USD]	
	COSTO INDIVIDUAL DE FOCO EN "N" HORAS	COSTO PARA "N" FOCOS	COSTO INDIVIDUAL DE FOCO EN "N" HORAS	COSTO PARA "N" FOCOS	COSTO INDIVIDUAL DE FOCO EN "N" HORAS	COSTO PARA "N" FOCOS
INCANDESCENTE	0,03	0,14	0,51	2,05	12,50	49,98
AHORRADOR	0,01	0,05	0,17	0,68	4,17	16,66
LED	0,00	0,02	0,07	0,27	1,67	6,66

Figura 59. Consumo promedio de 6 horas diarias en dólares sin intervención del grupo de trabajo.

Elaborado por: Juan Fonseca y Daniel Soria

Así pues, se calcularon dos criterios para el día, mes y año, estos son: el precio que genera un foco durante un cierto número de horas y el precio que se genera por cierta cantidad de focos durante un periodo establecido obtenidos de la Figura 59. En tal sentido, se aplicó la siguiente fórmula en la Figura 60:

Precio unitario por día

$$PRECIO\ UNITARIO = CONSUMO \left[\frac{KWh}{día} \right] * COSTO\ KWh[USD]$$

Figura 60. Ecuación para calcular precio unitario en un día.

Elaborado por: Juan Fonseca y Daniel Soria

La Figura 59, demuestra los cálculos realizados para las siguientes variables:

- **Costo individual de foco en "n" horas en un día:** Consumo diario de un foco.
- **Costo para "n" focos:** Consumo diario de “n” focos.
- **Costo individual de foco en "n" horas:** Consumo mensual de un foco.
- **Costo para "n" focos:** Consumo mensual de “n” focos.
- **Costo individual de foco en "n" horas:** Consumo anual de un foco.
- **Costo para "n" focos:** Consumo anual de “n” focos

De las evidencias anteriores, se hizo hincapié en las siguientes cifras:

- 0,17 [USD], resultado del consumo de 1 foco ahorrador durante 6 horas diarias promedio en 15 días.
- 4,17 [USD], resultado del consumo de 1 foco ahorrador durante 6 horas diarias promedio en 1 año.
- 0,68 [USD], resultado del consumo de 4 focos ahorradores durante 6 horas diarias en 15 días.
- 16,66 [USD], resultado del consumo de 4 focos ahorradores durante 6 horas diarias en 1 año.

Finalmente, se procedió a calcular los tipos de consumo en potencia y en dólares basándose en la “PRUEBA B” con un promedio de 3,4 horas diarias perteneciente a la Figura 54, es decir, con la intervención del grupo de trabajo.

Para la Figura 61, se accedió a la misma investigación realizada en la Figura 55 para asignación de variables.

Variables para consumo energético

VARIABLES	
NÚMERO DE DÍAS AL AÑO:	365
NÚMERO DE DÍAS AL MES:	15
NÚMERO DE HORAS AL DÍA:	3,4
NÚMERO DE FOCOS:	4
COSTO USD/KWh	0,0951

Figura 61. Variables para cálculo de consumo de energía eléctrica en prueba B.
Elaborado por: Juan Fonseca y Daniel Soria

Consumo energético diario para un foco

CÁLCULO DE CONSUMO DE FOCOS			
TIPO DE FOCO	POTENCIA [W]	CONSUMO AL DÍA [W]	[KWh / día]
INCANDESCENTE	60	204	0,204
AHORRADOR	20	68	0,068
LED	8	27,2	0,0272

Figura 62. Cálculo de consumo diario de energía para un foco
Elaborado por: Juan Fonseca y Daniel Soria

Consumo energético mensual para un foco

CÁLCULO DE CONSUMO DE FOCOS			
TIPO DE FOCO	POTENCIA [W]	CONSUMO AL MES [W]	[KWh / mes]
INCANDESCENTE	60	3060	3,06
AHORRADOR	20	1020	1,02
LED	8	408	0,408

Figura 63. Cálculo de consumo mensual de energía para un foco
Elaborado por: Juan Fonseca y Daniel Soria

Consumo energético anual para un foco

CÁLCULO DE CONSUMO DE FOCOS			
TIPO DE FOCO	POTENCIA [W]	CONSUMO AL AÑO [W]	[KWh / año]
INCANDESCENTE	60	74460	74,46
AHORRADOR	20	24820	24,82
LED	8	9928	9,928

Figura 64. Cálculo de consumo anual de energía para un foco
Elaborado por: Juan Fonseca y Daniel Soria

Las Figuras 62, 63 y 64, exponen las tablas con sus cálculos respectivos para obtener el consumo de energía diaria, mensual y anual con un foco, especificando el tipo y potencia del mismo; por lo tanto, se utilizó la ecuación presentada en la Figura 53.

Costos de uno y “n” focos en un día, 15 días y un año

TIPO DE FOCO	AL DÍA [USD]		AL MES [USD]		AL AÑO [USD]	
	COSTO INDIVIDUAL DE FOCO EN "N" HORAS	COSTO PARA "N" FOCOS	COSTO INDIVIDUAL DE FOCO EN "N" HORAS	COSTO PARA "N" FOCOS	COSTO INDIVIDUAL DE FOCO EN "N" HORAS	COSTO PARA "N" FOCOS
INCANDESCENTE	0,02	0,08	0,29	1,16	7,08	28,32
AHORRADOR	0,01	0,03	0,10	0,39	2,36	9,44
LED	0,00	0,01	0,04	0,16	0,94	3,78

Figura 65. Consumo promedio de 3,4 horas diarias en dólares con intervención del grupo de trabajo.
Elaborado por: Juan Fonseca y Daniel Soria

Para obtener la información de la Figura 65, se utilizó la fórmula mostrada en la Figura 60, haciendo hincapié en las siguientes cifras:

- 0,10 [USD], resultado del consumo de 1 foco ahorrador durante 3,4 horas diarias promedio en 15 días.
- 2,36 [USD], resultado del consumo de 1 foco ahorrador durante 3,4 horas diarias promedio en un año.
- 0,39 [USD], resultado del consumo de 4 focos ahorradores durante 3,4 horas diarias en 15 días.
- 9,44 [USD], resultado del consumo de 4 focos ahorradores durante 3,4 horas diarias en un año.

VALORES COMPARADOS SIN Y CON INTERVENCIÓN DEL PRODUCTO

Tabla 78. Comparación de resultados para consumo energético durante 15 días.

	Costos sin intervención del grupo de trabajo	Costos con intervención del grupo de trabajo	Ahorro establecido
1 FOCO	0,17	0,10	0,07
4 FOCOS	0,68	0,39	0,29

Nota: Esta tabla contiene los datos relevantes del análisis de consumo energético durante 15 días.

Elaborado por: Juan Fonseca y Daniel Soria

En virtud de los resultados, se demostró en la Tabla 78, que existe un ahorro de 0,07 [USD] para un foco durante un periodo de 15 días, a su vez, existió un ahorro de 0,29 [USD] para 4 focos durante 15 días.

Tabla 79. Comparación de resultados para consumo energético durante 1 año.

	Costos sin intervención del grupo de trabajo	Costos con intervención del grupo de trabajo	Ahorro establecido
1 FOCO	4,17	2,36	1,81
4 FOCOS	16,66	9,44	7.22

Nota: Esta tabla contiene los datos relevantes del análisis de consumo energético durante 1 año.

Elaborado por: Juan Fonseca y Daniel Soria

En virtud de los resultados, se demostró en la Tabla 79, que existe un ahorro de 1,81 [USD] para un foco durante 15 días, y a su vez, existió un ahorro de 7,22 [USD] para 4 focos durante un año. Además, se comparó el precio del consumo anual para 4 focos ahorradores entre las Figuras 59 y 65, obteniendo un 43,33% de ahorro energético con la intervención del grupo de trabajo.

4.6. Prototipo Experimental

Concluido todo el proceso anterior, se verificó el producto final mediante la fase 7 de la metodología aplicada, mismo que se elaboró en un prototipo experimental con la arquitectura de un departamento de un piso representados en la Figura 2.

Prototipo experimental – vista externa



Figura 66. Vista externa del prototipo experimental con un sistema domótico.

Elaborado por: Juan Fonseca y Daniel Soria

Prototipo experimental – vista lateral

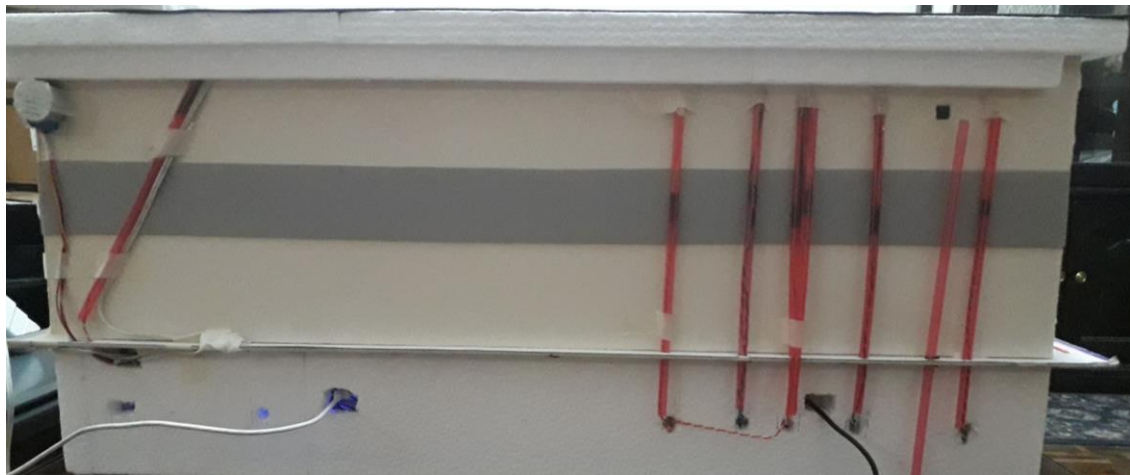


Figura 67. Vista lateral del prototipo experimental con un sistema domótico.
Elaborado por: Juan Fonseca y Daniel Soria

Prototipo experimental – vista interna



Figura 68. Vista interna del prototipo experimental con un sistema domótico.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 66 y Figura 67, muestran el prototipo experimental desde una vista frontal y lateral representando la estética propia de un hogar. Por otra parte, una vez implementados todos los

elementos desarrollados en este proyecto técnico, en la Figura 68, se visualizan los elementos colocados estratégicamente en el interior del modelo utilizado, con el objetivo de cumplir con el confort del usuario, además, se simuló un escenario real donde solo sean visibles sensores y actuadores, sin cables.

Prototipo experimental – diseño de circuitos

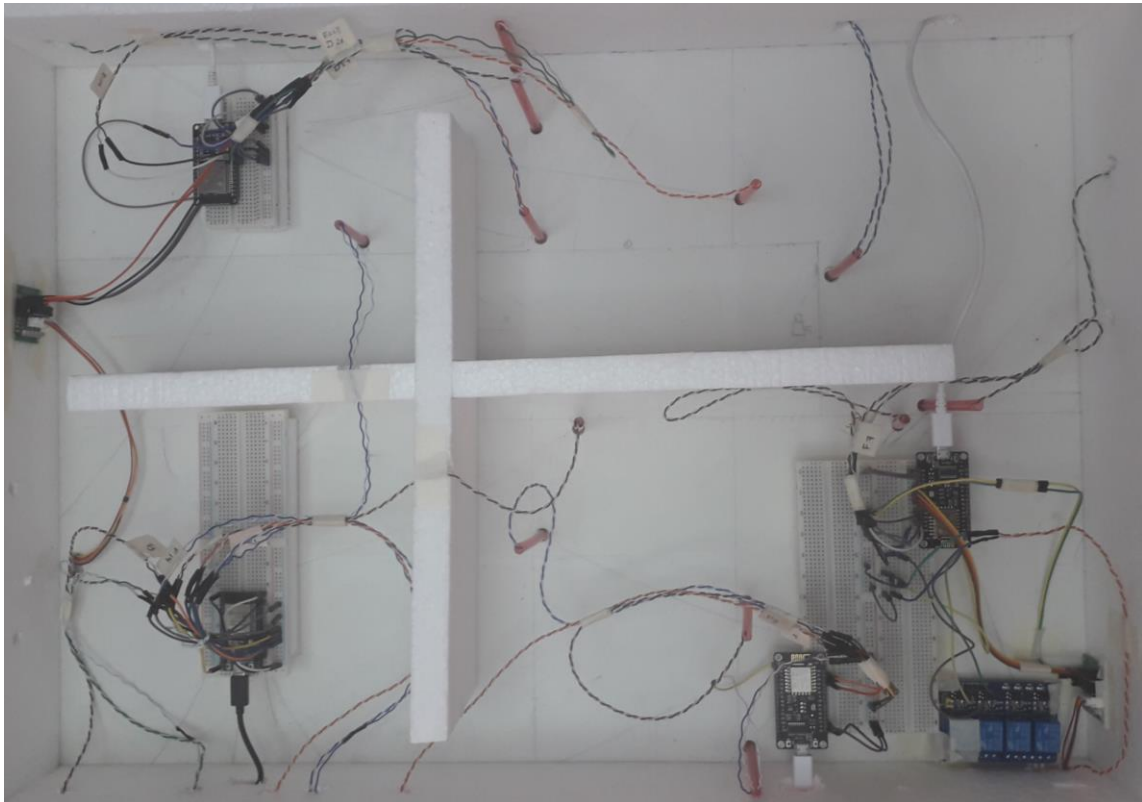


Figura 69. Cableado implementado en el prototipo experimental con un sistema domótico.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 69, presenta la elaboración e implementación de los circuitos pertenecientes a cada módulo descrito en el capítulo 3, para esto, se colocó en la parte inferior del prototipo dos placas ESP-8266 y dos placas ESP-32, mismas que fueron distribuidas con su respectivo etiquetado para iluminación artificial del prototipo y finalmente, se simuló el uso de canaletas con sorbetes adecuados a las longitudes de cada cable empleado. Finalmente, se enfatiza que las características para la instalación del producto en un escenario real varían, porque las placas serían implementadas en la pared del módulo a automatizar.

4.7. Vinculación de Hardware con Software

4.7.1. Encendido de focos (actuador) entre hardware y software.

En última instancia, se vincularon todos los elementos de este producto de hardware con el producto de software para su presentación definitiva.

Encendido de 4 focos desde la aplicación web hacia el hardware

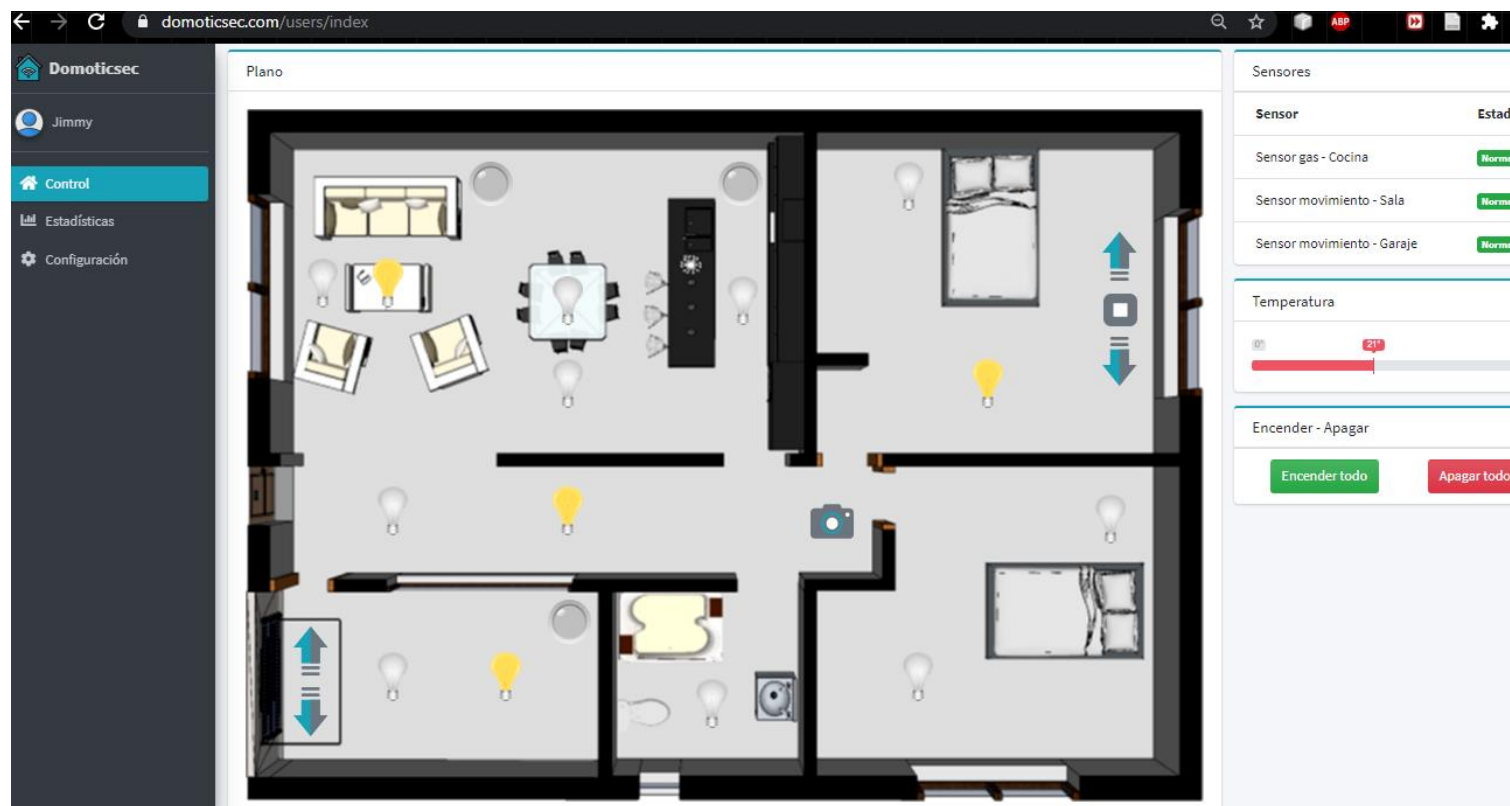


Figura 70. Envío de orden para encender focos desde la aplicación web con protocolo MQTT para su comunicación.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 70, presenta el envío de una orden desde la aplicación web por medio de su interfaz donde se puede interactuar con los elementos de hardware dispuestos en el prototipo experimental.

Para este literal se utilizó se aplicó la conectividad “M2M” desde el servidor al cliente, es decir, se realizó peticiones desde la aplicación web hacia los clientes ESP.

Encendido de cuatro focos en el prototipo experimental



Figura 71. Encendido de focos en el prototipo experimental con un sistema domótico.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 71, confirma la respuesta ejecutada por el hardware al momento de recibir una orden por parte de la aplicación web mediante la comunicación del protocolo MQTT.

4.7.2. Apertura y cierre de persiana (actuador) entre hardware y software.

Desde el literal 4.7.2. hasta 4.7.6. se aplicó la conectividad “M2M” desde el cliente al servidor, por lo tanto, se envió peticiones desde los clientes ESP hacia el servidor conjuntamente con la aplicación web.

Apertura y cierre de motor en la aplicación web

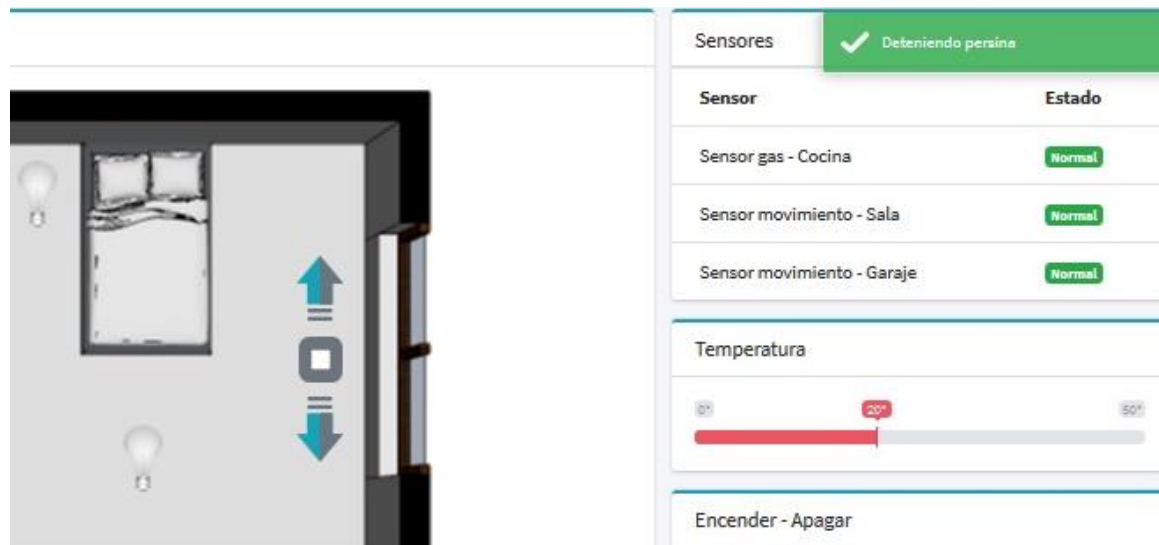


Figura 72. Visualización para cierre de persiana en la aplicación web.
Elaborado por: Juan Fonseca y Daniel Soria

Apertura y cierre de motor enviado desde el hardware

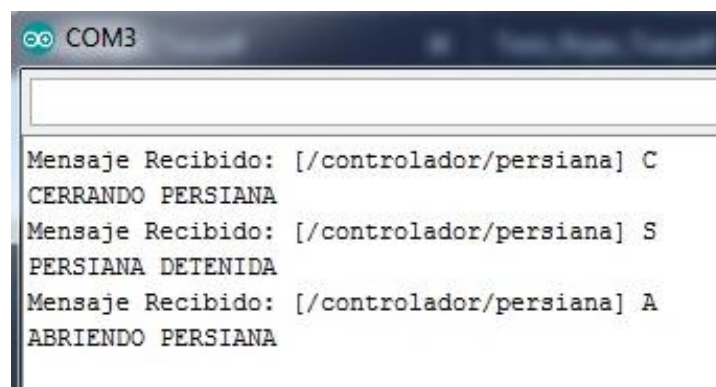


Figura 73. Resultado de orden en persiana desde el cliente hardware.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 72, muestra la orden entregada por el cliente hardware en interfaz de la aplicación web para abrir y cerrar una persiana, del mismo modo, en la Figura 73, se muestra el resultado en consola del envío de apertura y cierre de esta, desde el cliente hardware. A su vez, la Figura 74, muestra de manera física la forma en que la persiana se levanta con un motor en el prototipo experimental.

Apertura y cierre de motor en el prototipo experimental

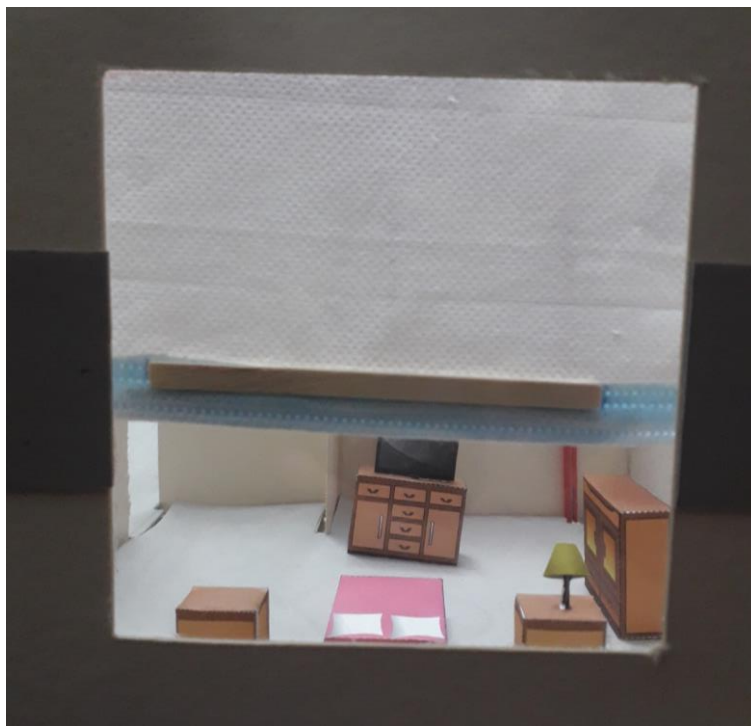


Figura 74. Apertura de persiana en el prototipo experimental.
Elaborado por: Juan Fonseca y Daniel Soria

4.7.3. Streaming y captura de imagen con placa ESP32-CAM desde el hardware.

Streaming con ESP32-CAM



Figura 75. Transmisión en tiempo real desde ESP32-CAM.
Elaborado por: Juan Fonseca y Daniel Soria

Captura de imagen con ESP32-CAM

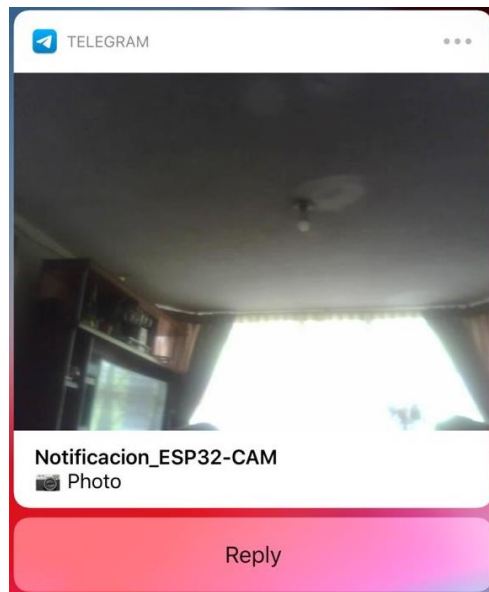


Figura 76. Alerta recibida en el dispositivo móvil del usuario desde ESP32-CAM.
Elaborado por: Juan Fonseca y Daniel Soria

Las Figuras 75 y 76, representan físicamente el proceso mencionado en el literal 3.2.4. del capítulo 3, a su vez, la Figura 77, muestra el resultado del envío de imágenes al usuario por consola.

Alerta y envío de imágenes con ESP32-CAM

```
...
STAIP address:
192.168.1.62

Intentando conexion MQTT...conectado
ESTADO NORMAL -0
- INTRUSO -
Connected to api.telegram.org
.....
{"ok":true,"result":{"message_id":7,"from":{"id":1077465405,"is_bot":true,"first_name":"Notificacion_ESP32-CAM","
  EXISTE MOVIMIENTO - IMAGEN ENVIADA A TELEGRAM
ESTADO NORMAL -0
- INTRUSO -
Connected to api.telegram.org
.....
{"ok":true,"result":{"message_id":8,"from":{"id":1077465405,"is_bot":true,"first_name":"Notificacion_ESP32-CAM","
  EXISTE MOVIMIENTO - IMAGEN ENVIADA A TELEGRAM
ESTADO NORMAL -0
- INTRUSO -
Connected to api.telegram.org
.....
```

Figura 77. Envío de imágenes en tiempo real desde el cliente hardware ESP32-CAM.
Elaborado por: Juan Fonseca y Daniel Soria

4.7.4. Alerta por detección de movimiento (sensor) entre hardware y software.

Alerta por detección de movimiento no autorizado en la aplicación web

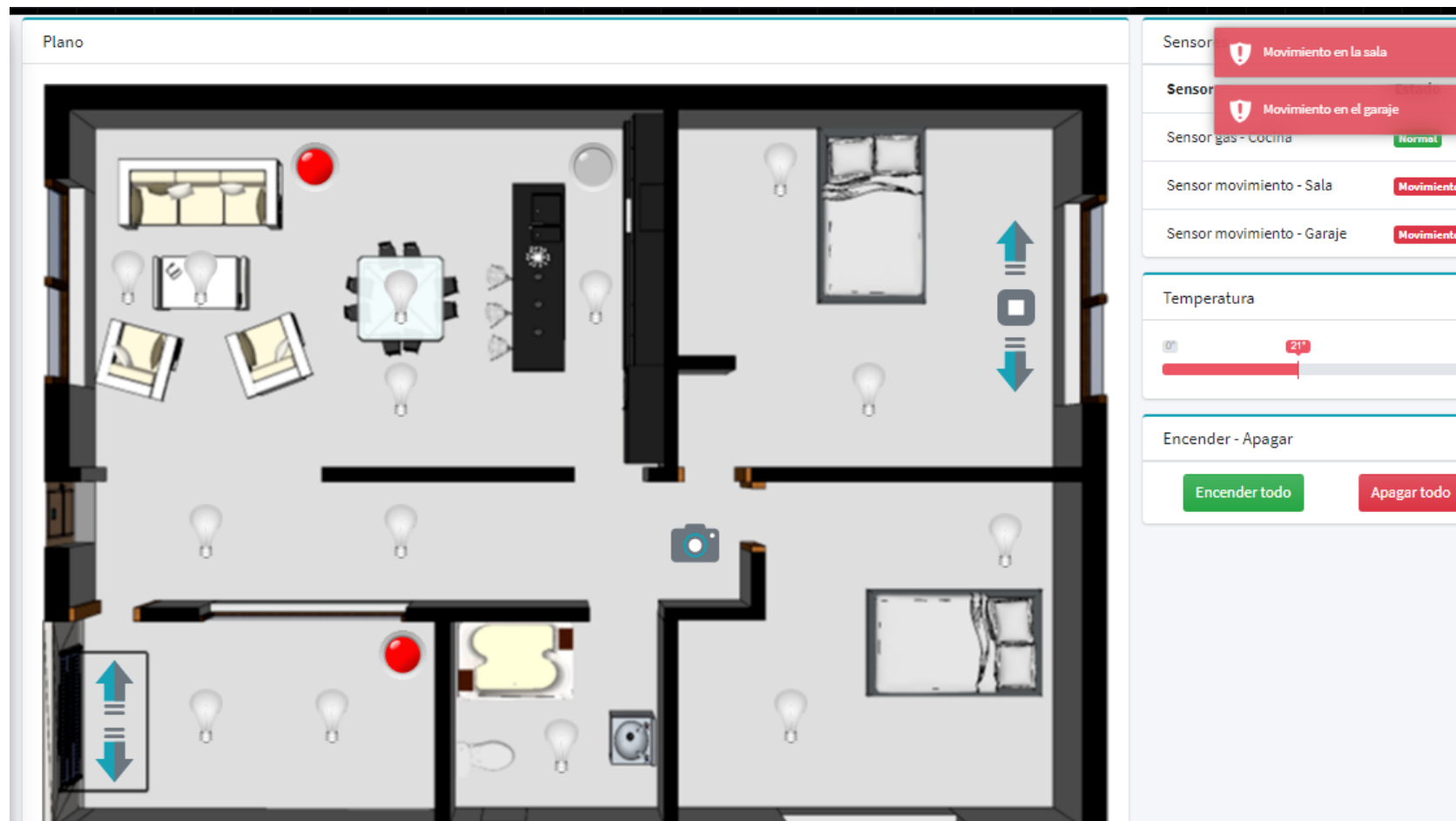


Figura 78. Alerta de movimiento no autorizado entregada por el sensor PIR a la interfaz de la aplicación web.
Elaborado por: Juan Fonseca y Daniel Soria

La Figura 79, notifica una alerta de movimiento no autorizada en la aplicación web, del mismo modo, la Figura 80, realiza el proceso anterior en la interfaz MQTT BROKER.

Alerta por detección de movimiento no autorizado en cloud MQTT

Received messages

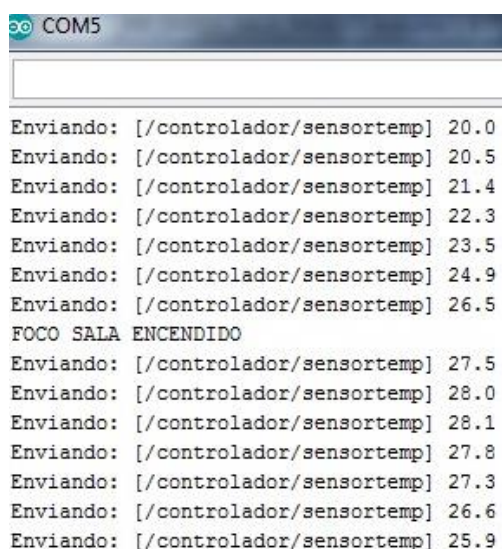
Topic	Message
/controlador/pirgarage	Movimiento
/controlador/ldr	OFF SALA APAGA
/controlador/sensortemp	20.5
/controlador/sensorpir	Movimiento
/controlador/pirgarage	Movimiento

Figura 79. Alerta de movimiento no autorizado en cloud MQTT enviado por el cliente hardware que cuenta con un sensor PIR.

Elaborado por: Juan Fonseca y Daniel Soria

4.7.5. Notificación de temperatura (sensor) entre hardware y software.

Variación de temperatura enviada desde el hardware



```
COM5
Enviando: [/controlador/sensortemp] 20.0
Enviando: [/controlador/sensortemp] 20.5
Enviando: [/controlador/sensortemp] 21.4
Enviando: [/controlador/sensortemp] 22.3
Enviando: [/controlador/sensortemp] 23.5
Enviando: [/controlador/sensortemp] 24.9
Enviando: [/controlador/sensortemp] 26.5
FOCO SALA ENCENDIDO
Enviando: [/controlador/sensortemp] 27.5
Enviando: [/controlador/sensortemp] 28.0
Enviando: [/controlador/sensortemp] 28.1
Enviando: [/controlador/sensortemp] 27.8
Enviando: [/controlador/sensortemp] 27.3
Enviando: [/controlador/sensortemp] 26.6
Enviando: [/controlador/sensortemp] 25.9
```

Figura 80. Envío de variación de temperatura desde el sensor DHT-11.

Elaborado por: Juan Fonseca y Daniel Soria

En este literal se muestra la variación de temperatura, donde la Figura 81, muestra el resultado en consola enviado por el sensor DHT-11. Mientras que la Figura 82, mediante una gráfica lineal junto a un indicador muestran la diversificación de temperatura.

Variación de temperatura recibida en la aplicación web

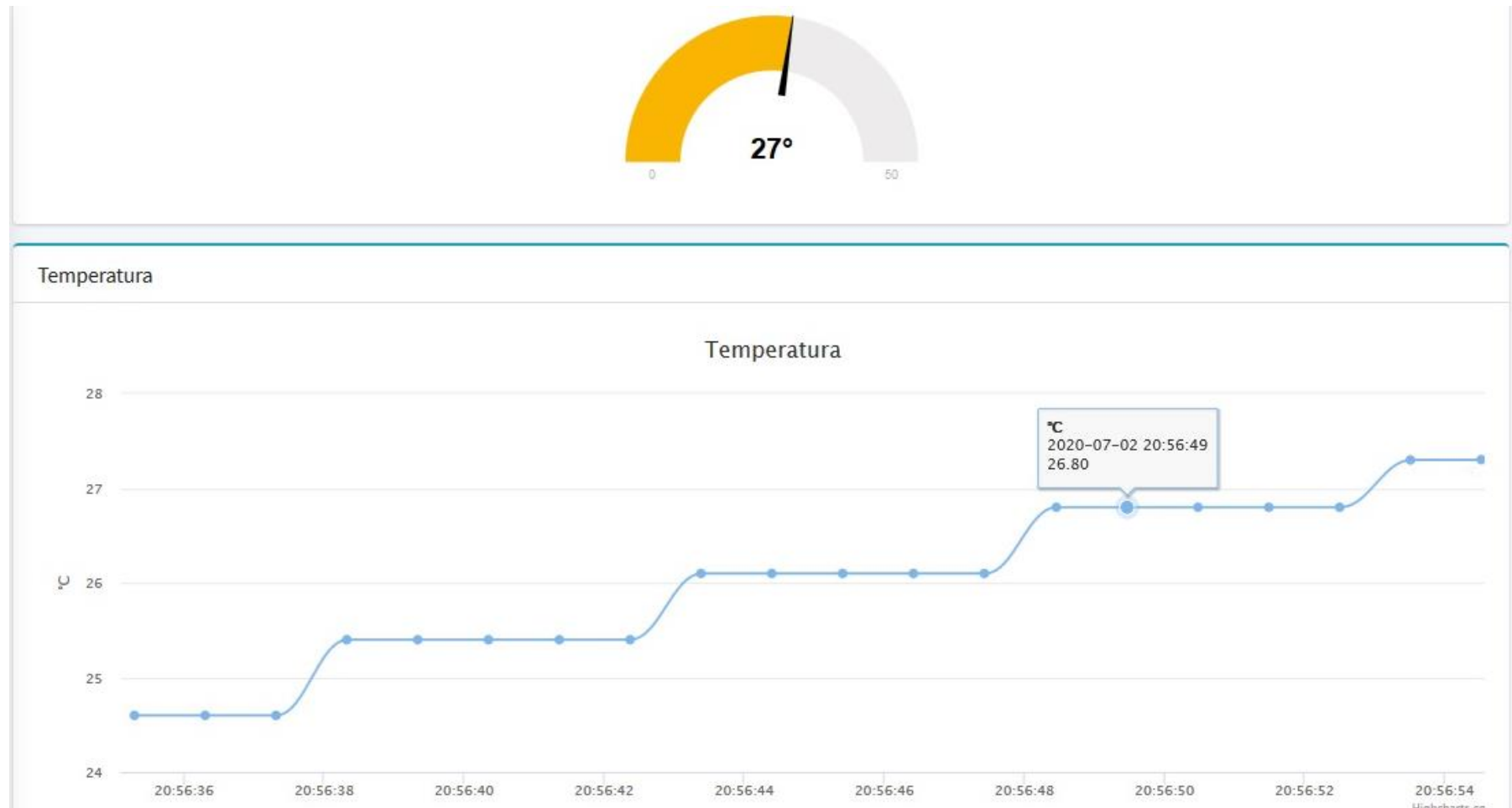


Figura 81. Variación de temperatura entregada por el cliente hardware en interfaz de la aplicación web.
Elaborado por: Juan Fonseca y Daniel Soria

4.7.6. Alerta por detección de gas (sensor) entre hardware y software.

La alerta mostrada por las Figuras 83 y 84, fue condicionada en caso de que, si el nivel de gas persiste sobre su nivel normal de ambiente, el hardware continuará alertando al usuario hasta que tome las debidas precauciones.

Alerta por detección de gas en la aplicación web

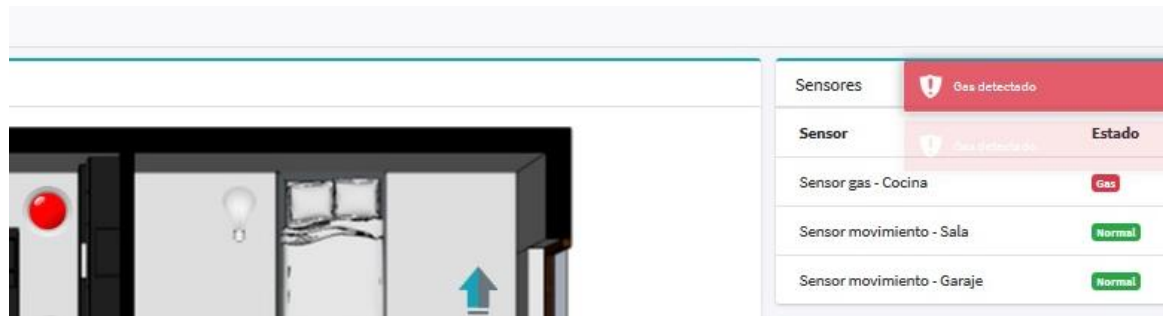


Figura 82. Alerta por detección de gas entregada por el cliente hardware en interfaz de la aplicación web
Elaborado por: Juan Fonseca y Daniel Soria

Alerta por detección de gas enviada desde el hardware

```
COM4
..Intentando conexion MQTT...conectado
Enviando: [/controlador/sensormq6] 93.0
Enviando: [/controlador/sensormq6] 94.0
Enviando: [/controlador/sensormq6] 90.0
Enviando: [/controlador/sensormq6] 148.0
```

Figura 83. Alerta por detección de gas enviada desde el cliente hardware que cuenta con un sensor MQ-6.
Elaborado por: Juan Fonseca y Daniel Soria

Finalmente, en el literal 4.7, se presentó el resultado del proyecto final vinculado entre hardware y software; teniendo en cuenta el trabajo sincronizado y continuo durante el desarrollo de ambos proyectos técnicos.

Conclusiones

- El uso del Modelo en V del Ciclo de Vida, permite realizar desarrollo y pruebas entre hardware y software definiendo especificaciones y requisitos, por esta razón, se elaboraron pruebas de tiempos de respuesta para cada protocolo investigado (HTTP, MySQL, WebSockets, MQTT-SN), siendo MQTT-SN la mejor opción como protocolo de transmisión de datos con el 96% de efectividad. Finalmente, al implementar el protocolo mencionado en el prototipo experimental, se obtuvo mediante un análisis final una efectividad del 97,14%.
- La implementación de este producto debió contemplar requisitos específicos como su protocolo de medio de comunicación domótico, protocolo de transmisión de datos, dominio y un ISP; por consiguiente, tras investigar y ejecutar pruebas de funcionalidad se definió que según la Figura 49, C-BUS fue idóneo debido a que su medio de transmisión se realiza vía inalámbrica (WiFi) y según la Tabla 74, MQTT-SN se destacó porque transmitió información entre gestor – suscriptor y viceversa con un promedio de retardos de 0,48 [s] en sus respuestas.
- El envío y recepción de datos realizados en el proyecto, contaron con un mecanismo de confiabilidad de QoS nivel uno (reenvío de mensajes hasta obtener confirmación por parte del receptor), permitiendo de esta manera que la información sea confiable.
- El consumo energético dentro de los hogares se incrementa por el uso incorrecto de elementos como luces, por este motivo, se desarrolló un sistema adecuado que administre la iluminación artificial interna del hogar, permitiendo al usuario final conocer el estado del foco (encendido/apagado) vinculado al concepto de simular todo en un prototipo experimental. Así pues, según la Tabla 80, se analizó y concluyó que con el uso de este prototipo se redujo un 43,33% el consumo energético e inclusive permitió ahorrar 1,81 [USD] en un foco ahorrador al año y 7,22 [USD] en cuatro focos ahorradores al año.

- Debido al incremento del 17% de inseguridad en Quito durante el 2019, junto al 52% de un grupo de personas encuestadas que desearían poseer un sistema de videovigilancia, dieron paso para desarrollar el proyecto técnico con el fin de incrementar la seguridad del domicilio con una cámara que transmitió video en tiempo real y otra cámara que capturó imágenes; en caso de recibir una alerta por parte de un sensor de movimiento se remitió la imagen vía mensaje multimedia (utilizando la aplicación Telegram) al celular del usuario administrador.

Recomendaciones

- Según el estudio realizado en este proyecto, para obtener un rendimiento adecuado se recomienda utilizar más de una tarjeta de desarrollo ESP puesto que cuentan con un procesador limitado para ejecución de órdenes.
- Las pruebas ejecutadas exigieron una conexión estable a internet que permita fluidez de transmisión de datos entre cliente y servidor, debido a esto, se recomienda al usuario final poseer un “ISP” que garantice un buen desempeño en su servicio.
- Se recomienda que los sensores de gas, temperatura/humedad y movimiento sean energizados a 5 voltios con un mínimo de 24 horas de anterioridad para disponer de una mejor precisión en los datos recolectados, debido a que, al consultar valores inmediatamente a su conexión, la información será irreal.
- Los pines GPIO (General Purpose Input/Output) de las tarjetas de desarrollo ESP-8266 y ESP-32 deben ser investigados en detalle puesto que cada uno cumple una función específica, debido a esto, si se conecta un sensor o actuador a un pin que cumple alguna función específica, no existirá respuesta por parte de la placa.

Lista De Referencias

Artículo de periódico

Grupo EL COMERCIO. (2020, febrero 14). En 2019, la inseguridad aumentó en un 17% en Quito. *El Comercio*. Recuperado de <http://www.elcomercio.com/actualidad/inseguridad-cifras-quito-operativos-policia.html>

Artículos académicos

Gubbi, J., Marusi, S., Buyya, R., & Palaniswam, M. (2013). *Internet of Things (IoT): A vision, architectural elements, and future directions*. 29(7), 1645-1660.

Medina Valle, I. G. (2012). *Sistema domótico con tecnología EIB KONNEX para la automatización de servicios, confort y seguridad en la Empresa Sisteldata S.A.* Recuperado de <https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/2339>

Pérez, A., Berreteaga, O., Ruiz de Olano, A., Urkidi, A., & Perez, J. (2006). UNA METODOLOGÍA PARA EL DESARROLLO DE HARDWARE Y SOFTWARE EMBEBIDOS EN SISTEMAS CRÍTICOS DE SEGURIDAD. *SISTEMAS, CIBERNÉTICA E INFORMÁTICA*, 3(2), 6.

Stanford-Clark, A., & Linh Truong, H. (2013). *MQTT For Sensor Networks (MQTT-SN)*. 28.

Artículos en conferencia

Ramírez, J., & Pedraza, C. (2017, octubre 30). *Performance analysis of communication protocols for Internet of things platforms*. Presentado en 2017 IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena, Colombia. <https://doi.org/10.1109/ColComCon.2017.8088198>

Documentos

AI-Thinker. (2015). *ESP-01 WiFi Module—Version 1.0*. Recuperado de <https://pdf1.alldatasheet.com/datasheet-pdf/view/1179098/ETC2/ESP-01.html>

AI-Thinker. (2017). *ESP32-CAM Product Specification*. Recuperado de <https://loboris.eu/ESP32/ESP32-CAM%20Product%20Specification.pdf>

Aosong Electronics Co. (s. f.). *Digital-output relative humidity & temperature sensor/module DHT22*. Recuperado de <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

ARCONEL. (2017). *PRECIO MEDIO DE ENERGÍA FACTURADA (USD ¢/kWh)*. Recuperado de https://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2017/10/15_PRECIO_MEDIO_ENERGIA_FACTURADA.pdf

- CEBEK. (s. f.). *FOTORRESISTENCIA LDR 4,3mm x Ø 5,1mm*. Recuperado de <https://www.electan.com/datasheets/cebek/CE-C2795.pdf>
- Espressif Inc. (2019b). *ESP32-LyraT Datasheet*. Recuperado de https://www.espressif.com/sites/default/files/documentation/esp32-lyrat_user_guide_en.pdf
- Espressif Inc. (2019c). *ESP32-WROOM-32 Datasheet*. Recuperado de https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- Espressif Inc. (2019d). *ESP8266EX Datasheet*. Recuperado de https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- Handson Technology. (2018). *HC-SR04 Ultrasonic Sensor Module*. Recuperado de <https://handsontec.com/index.php/product/hc-sr04-ultrasonic-ranging-module/>
- Handson Technology. (s. f.). *ESP8266 NodeMCU WiFi Devkit*. Recuperado de https://www.handsontec.com/pdf_learn/esp8266-V10.pdf
- Mouser Electronics. (2019). *DHT11 Humidity & Temperature Sensor*. Recuperado de <https://www.mouser.ec/Search/Refine?N=4248457961&Keyword=dht11>
- Olimex. (2019). *MQ-2 Semiconductor Sensor for Combustible Gas*. Recuperado de <https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ2/>
- Punto Flotante S.A. (2017). *MANUAL DEL USUARIO - SENSOR INFRARROJO DE MOVIMIENTO PIR HC-SR501*. Recuperado de <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>
- ST Engineering. (2018). *HB 100 Microwave Sensor Datasheet*. Recuperado de <https://agilsense.com/product/detail/3>
- Zhengzhou Winsen Electronics Technology CO., LTD. (s. f.). *MQ-6 Semiconductor Sensor for Flammable Gas*. Recuperado de <https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-6%20Ver1.3%20-%20Manual.pdf>

Informe

- Instituto Nacional de Estadística y Censos (INEC). (2016). *Tecnologías de la Información y Comunicaciones (TIC'S) 2016* (p. 39) [Estadístico_Social]. Ecuador. Recuperado de https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2016/170125.Presentacion_Tics_2016.pdf

Libros

- Germán Corona Ramírez, L., Abarca Jiménez, G. S., & Mares Carreño, J. (2014). *Sensores y actuadores: Aplicaciones con Arduino*. Distrito Federal, UNKNOWN: Grupo Editorial Patria. Recuperado de <http://ebookcentral.proquest.com/lib/bibliotecaupssp/detail.action?docID=4569609>
- Junestrand, S., Passaret, X., & Vázquez, D. (2004). *Domótica y Hogar Digital*. Madrid: Editorial Paraninfo. Recuperado de https://books.google.com.ec/books?id=8ERFqWcdHAEC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false
- Ruiz Buitrón, C. E. (2020). *Sistemas integrados y hogar digital* (Primera). España: Ediciones Paraninfo, S.A. Recuperado de <https://books.google.com.ec/books?id=0AHQDwAAQBAJ&pg=PA85&lpg=PA85&dq=velocidad+de+comunicacion+en+tcp/ip+para+domotica&source=bl&ots=Yvwt8Wgjh&sig=ACfU3U1jAYp5qm6TU2khu3jCGit4QFDuBA&hl=es-419&sa=X&ved=2ahUKEwiXo6PEj4rqAhWEiOAKHaWBCzUQ6AEwA3oECAoQAQ#v=onepage&q=velocidad%20de%20comunicacion%20en%20tcp%2Fip%20para%20domotica&f=false>

Sitios web

- Arduino. (2015). Arduino—Environment. Recuperado 14 de diciembre de 2019, de <https://www.arduino.cc/en/Guide/Environment#>
- Arduino. (2020). Arduino Create. Recuperado 30 de enero de 2020, de Arduino Project Hub website: https://create.arduino.cc/projecthub/Arduino_Genuino/act-on-multiple-sketches-at-once-on-the-web-editor-4cdf07
- Brandão, R. (2019, enero 6). ESP32-CAM_Product_Specification. Recuperado 18 de diciembre de 2019, de GitHub website: <https://github.com/raphaelbs/esp32-cam-ai-thinker>
- CENTROSUR. (2018, septiembre 26). Calcular consumo de energía eléctrica [Empresa Eléctrica Regional Centro Sur C.A.]. Recuperado 3 de julio de 2020, de Centrosur website: <https://www.centrosur.gob.ec/calcular-consumo/>
- COMPONENTS101. (2017). Servomotor SG-90, descripción del cable, hoja de datos. Recuperado 17 de enero de 2020, de Servomotor SG-90 website: <https://components101.com/servo-motor-basics-pinout-datasheet>
- COMPONENTS101. (2018). Toy/Hobby DC Motor Pinout Wiring, Specifications, Uses Guide and Datasheet. Recuperado 17 de enero de 2020, de Toy DC Motor website: <https://components101.com/motors/toy-dc-motor>
- esp8266. (2014). ESPlorer IDE. Recuperado 30 de enero de 2020, de Esp8266 website: <https://esp8266.ru/esplorer/>

Espressif Inc. (2019a). ESP32-DevKitC V4 Getting Started Guide. Recuperado 4 de enero de 2020, de ESP32-DevKitC V4 Getting Started Guide website: <https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/get-started-devkitc.html>

Espressif Inc. (2019e). ESP-EYE. Recuperado 8 de marzo de 2020, de ESP-EYE website: <https://www.espressif.com/en/products/hardware/esp-eye/overview>

Espressif Inc. (2019f). ESP-WROVER-KIT V4.1 Getting Started Guide. Recuperado 3 de enero de 2020, de ESP-WROVER-KIT V4.1 Getting Started Guide website: <https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/get-started-wrover-kit.html>

Llamas, L. (2018, septiembre 22). Wemos D1 Mini, una genial placa de desarrollo con ESP8266. Recuperado 24 de diciembre de 2019, de ESP8266/ESP32 —TUTORIALES ARDUINO website: <https://www.luisllamas.es/wemos-d1-mini-una-genial-placa-de-desarrollo-con-esp8266/>

Ministerio de Energía y Recursos Naturales no Renovables. (2020, enero 9). En Ecuador, el consumo de energía eléctrica se incrementó en un 4,5% en 2019 [Boletín de Prensa]. Recuperado 29 de julio de 2020, de Ministerio de Energía y Recursos Naturales no Renovables website: <https://www.rekursyenergia.gob.ec/en-ecuador-el-consumo-de-energia-electrica-se-incremento-en-un-45-en-2019/>

PHP Group. (s. f.). PHP: ¿Qué es PHP? - Manual. Recuperado 14 de diciembre de 2019, de ¿Qué es PHP? website: <https://www.php.net/manual/es/intro-what-is.php>

Promotec. (s. f.). Motores paso a paso: 28BYJ-48 [MOTORES PASO A PASO: 28BYJ-48]. Recuperado 16 de enero de 2020, de <https://www.promotec.net/motor-28byj-48/>

ROHM CO., LTD. (s. f.). Light Emitting Diode <What are LEDs and How Do They Work?>. Recuperado 14 de diciembre de 2019, de ROHM SEMICONDUCTOR website: <https://www.rohm.com/electronics-basics/leds/what-are-leds>

Waveshare. (2019). MQ-5 Gas Sensor. Recuperado 16 de enero de 2020, de MQ-5 Gas Sensor website: <https://www.waveshare.com/mq-5-gas-sensor.htm>

Zerynth. (s. f.). Wemos D1 Mini—Zerynth Docs documentation. Recuperado 24 de diciembre de 2019, de Wemos D1 Mini website: https://docs.zerynth.com/latest/official/board.zerynth.wemos_d1_mini/docs/index.html

Tesis

Camargo, C. (2011). *Transferencia tecnológica y de conocimientos en el diseño de sistemas embebidos* (Universidad Nacional de Colombia). Universidad Nacional de Colombia, Bogota, Colombia. Recuperado de <http://bdigital.unal.edu.co/5696/>

- Fernández Aragón, I. (2011). *Control de un motor paso a paso: PIC, USB, C#* (Proyecto Fin de Carrera, Universidad Pública de Navarra.). Universidad Pública de Navarra., Navarra. Recuperado de <https://academica-e.unavarra.es/xmlui/handle/2454/3547>
- Herranz, Á. (2019). *Desarrollo de aplicaciones para IoT con el módulo ESP32* (Grado en Ingeniería en Tecnologías de Telecomunicación, Universidad de Alcalá Escuela Politécnica Superior). Universidad de Alcalá Escuela Politécnica Superior, Madrid, España. Recuperado de https://ebuah.uah.es/dspace/bitstream/handle/10017/35420/TFG_Benito_Herranz_2019.pdf?sequence=1&isAllowed=y
- Hervas Parra, C. (2018). *Análisis de rendimiento de protocolos de publicación/subscripción en comunicación con una red de sensores inalámbricos Zigbee* (Magister en Redes de Datos, Universidad Nacional de La Plata). Universidad Nacional de La Plata. <https://doi.org/10.35537/10915/69435>
- Moreno Vargas, G. A., & Valdez Bravo, G. I. (2015). *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA EL MONITOREO EN TIEMPO REAL DE CCTV PARA DISPOSITIVOS ANDROID, HACIENDO USO DE LA RED CELULAR* (Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS, Universidad Politécnica Salesiana). Universidad Politécnica Salesiana, Guayaquil, Ecuador. Recuperado de <http://dspace.ups.edu.ec/handle/123456789/10367>
- Rosero, D. (2016). *SISTEMA EMBEBIDO PARA SUPERVISIÓN Y ADQUISICIÓN DE DATOS DE UN CONTROL DOMÓTICO*. (Universidad De San Buenaventura). Universidad De San Buenaventura, Cali, Colombia. Recuperado de http://bibliotecadigital.usbcali.edu.co/bitstream/10819/4264/1/Sistema_embebido_supervision_rosero_2016.pdf
- Ulloa Arízaga, E. S. (2015). *EFICIENCIA DEL CONSUMO ELECTRICO EN EL SECTOR RESIDENCIAL URBANO DE CUENCA* (MAESTRÍA EN PLANIFICACIÓN Y GESTIÓN ENERGÉTICA, Universidad de Cuenca). Universidad de Cuenca, Cuenca - Ecuador. Recuperado de <http://dspace.ucuenca.edu.ec/handle/123456789/22992>